

### Линейная последовательность действий.

1. Даны два ненулевых числа. Найти их сумму, разность, произведение, частное, остаток и целую часть от деления.
2. Даны два числа. Найти среднее арифметическое их квадратов и среднее арифметическое их модулей.
3. Скорость лодки в стоячей воде  $V$  км/ч, скорость течения реки  $U$  км/ч ( $U < V$ ). Время движения лодки по озеру  $T_1$  ч, а по реке (против течения) —  $T_2$  ч. Определить путь  $S$ , пройденный лодкой.
4. Скорость первого автомобиля  $V_1$  км/ч, второго —  $V_2$  км/ч, расстояние между ними  $S$  км. Определить расстояние между ними через  $T$  часов, если автомобили первоначально движутся навстречу друг другу.
5. Найти периметр и площадь прямоугольного треугольника, если даны длины его катетов  $a$  и  $b$ .
6. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
7. Найти длину окружности и площадь круга заданного радиуса  $R$ . В качестве значения  $P_i$  использовать 3.14.
8. Дана сторона равностороннего треугольника. Найти площадь этого треугольника и радиусы вписанной и описанной окружностей.
9. Дана длина окружности. Найти площадь круга, ограниченного этой окружностью. В качестве значения  $P_i$  использовать 3.14.
10. Найти периметр и площадь равнобедренной трапеции с основаниями  $a$  и  $b$  ( $a > b$ ) и углом  $\alpha$  при большем основании (угол дан в радианах).
11. Найти периметр и площадь прямоугольной трапеции с основаниями  $a$  и  $b$  ( $a > b$ ) и острым углом  $\alpha$  (угол дан в радианах).
12. Найти расстояние между двумя точками с заданными координатами  $(x_1, y_1)$  и  $(x_2, y_2)$ .
13. Даны координаты трех вершин треугольника  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Найти его периметр и площадь.
14. Найти корни квадратного уравнения  $Ax^2 + Bx + C = 0$ , заданного своими коэффициентами  $A$ ,  $B$ ,  $C$  (коэффициент  $A$  не равен 0), если известно, что дискриминант уравнения неотрицателен.
15. Найти решение системы уравнений вида
$$\begin{aligned}A_1x + B_1y &= C_1, \\ A_2x + B_2y &= C_2,\end{aligned}$$
заданной своими коэффициентами  $A_1$ ,  $B_1$ ,  $C_1$ ,  $A_2$ ,  $B_2$ ,  $C_2$ , если известно, что данная система имеет единственное решение.
16. Дано целое шестизначное число. Используя операции `div` и `mod`, найти сумму его цифр.
17. Поменяйте между собой значения двух переменных  $a$  и  $b$ , не используя третьей переменной.
18. Поменяйте между собой значения трех переменных  $a$ ,  $b$  и  $c$  по схеме тройного квартирного обмена:
$$a \rightarrow b \rightarrow c \rightarrow a$$

## Условия.

1. Ввести 4 оценки в баллах за экзамен (0-100). Вводимые баллы не должны быть отрицательными. Посчитать среднее количество баллов. Напечатать соответствующее сообщение в зависимости от среднего балла: Если средний балл ниже 60 – «не сдано», при среднем балле между 60 и 70 вывести «сдано посредственно», а если оценка не меньше 70 – «сдано». Написать данные – тесты для полного тестирования программы.

2. Тангенс угла наклона прямой, проходящей через точки с координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$  считается по формуле

$$tg(\alpha) = \frac{y_1 - y_2}{x_1 - x_2}$$

Если  $x_1 = x_2$ , то прямая вертикальна, и тангенс не определен (на 0 делить нельзя). Посчитать и напечатать тангенс угла наклона прямой по заданным координатам  $x_1, y_1, x_2, y_2$ . Если тангенс не существует выдать соответствующее сообщение.

3. Определить, верно ли, что при делении неотрицательного целого числа  $a$  на положительное целое число  $b$  получается остаток, равный одному из двух заданных чисел  $z$  или  $s$ . Числа  $a, b, z, s$  вводятся пользователем.
4. Дано натуральное число  $n$  ( $n \leq 99$ ). Выяснить, верно ли, что  $n^2$  равно кубу суммы цифр числа  $n$ .
5. Дано натуральное число  $n$  ( $n \leq 9999$ ). Является ли это число палиндромом (перевертышем) с учетом четырех цифр, как, например, числа 2222, 6116, 0440 и т. д.?
6. Дано натуральное число  $n$  ( $n \leq 9999$ ). Верно ли, что это число содержит ровно три одинаковые цифры, как, например, числа 6676, 4544, 0006 и т. д.?
7. Даны натуральные числа  $n, m$ . Найти произведение этих чисел если они являются удвоенными нечетными числами. Если условие не выполняется поменять их значения.
8. Даны целые числа  $n, m, p, q$ . Заменить нулями  $n$  и  $m$ , если модуль этих чисел при делении на  $p$  дает остаток  $q$ .
9. Дано целое число  $x$ . найти значение  $y$ , где  
 $y = x^2$ , если  $x$  кратно 3  
 $y = x^3$ , если  $x$  при делении на 3 дает остаток 1.  
 $y = x/2$  в остальных случаях.
10. Дано целое число  $i$ . найти значение  $a$ , где  
 $a = i^3$ , если  $i$  кратно 5  
 $a = i^2$ , если  $i$  кратно 2.  
 $a = 2*i + 1$  в остальных случаях.
11. Даны целые числа  $n, m, p$ . Нечетные из этих чисел заменить на удвоенные, четные числа уменьшить на 1.
12. Даны вещественные числа  $n, m, p$ . Найти квадрат максимального из трех чисел.
13. Дано натуральное  $n$  ( $n \leq 9999$ ). Выбросить из записи целого числа цифры 0 и 5, оставив прежний порядок остальных цифр. Если все цифры числа подлежат отбрасыванию, тогда заменить его на -1. Например, из числа 5905 должно получиться 9, из числа 5055 должно получиться -1.
14. Даны вещественные числа  $n, m, p$ . Если они упорядочены по возрастанию, заменить их квадратами. Если упорядочены по убыванию, увеличить в 2 раза, в противном случае уменьшить в 2 раза.
15. Даны действительные числа  $n, m, p$ . Выбрать из них те, которые принадлежат интервалу  $(0; 3.5)$ .
16. Даны действительные попарно различные числа  $n, m, p$ . Меньшее из этих трех чисел заменить полусуммой, а большее удвоенным произведением этих чисел.
17. Если сумма трех попарно различных чисел  $x, y, z$  меньше 1, то наименьшее из этих трех заменить полусуммой двух других; в противном случае заменить наибольшее полусуммой двух оставшихся.
18. Даны действительные числа  $x, y$ . Если  $x$  и  $y$  отрицательны, то каждое значение заменить его модулем; если отрицательно только одно из них, то оба значения увеличить на 0.5; в остальных случаях  $x$  и  $y$  оставить без изменения.

19. Дано натуральное число  $n$  ( $n \leq 100000$ ), определяющее возраст предмета (в годах). Дать для этого числа наименования «год», «года», «лет»: например, 1 год, 23 года, 45 лет.
20. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.).
21. Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней в этом месяце для невисокосного года.
22. Дано целое число в диапазоне 0 — 9. Вывести строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два", ...).
23. Дано целое число в диапазоне 1 — 5. Вывести строку — словесное описание соответствующей оценки (1 — "плохо", 2 — "неудовлетворительно", 3 — "удовлетворительно", 4 — "хорошо", 5 — "отлично").
24. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа А и В (В не равно нулю). Выполнить над числами указанное действие и вывести результат.
25. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах.
26. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы и масса тела M в этих единицах (вещественное число). Вывести массу данного тела в килограммах.
27. Робот может перемещаться в четырех направлениях ("С" — север, "З" — запад, "Ю" — юг, "В" — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, -1 — поворот направо. Дан символ С — исходное направление робота и число N — посланная ему команда. Вывести направление робота после выполнения полученной команды.
28. Локатор ориентирован на одну из сторон света ("С" — север, "З" — запад, "Ю" — юг, "В" — восток) и может принимать три цифровые команды: 1 — поворот налево, -1 — поворот направо, 2 — поворот на 180 градусов. Дан символ С — исходная ориентация локатора и числа N1 и N2 — две посланные ему команды. Вывести ориентацию локатора после выполнения данных команд.
29. Элементы окружности пронумерованы следующим образом: 1 — радиус (R), 2 — диаметр (D), 3 — длина (L), 4 — площадь круга (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке). В качестве значения  $P_i$  использовать 3.14.
30. Элементы равнобедренного прямоугольного треугольника пронумерованы следующим образом: 1 — катет (a), 2 — гипотенуза (c), 3 — высота, опущенная на гипотенузу (h), 4 — площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).
31. Элементы равностороннего треугольника пронумерованы следующим образом: 1 — сторона (a), 2 — радиус вписанной окружности (R1), 3 — радиус описанной окружности (R2), 4 — площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).
32. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, предшествующей указанной.
33. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, следующей за указанной.
34. Дано целое число в диапазоне 20 — 69, определяющее возраст (в годах). Вывести строку — словесное описание указанного возраста, обеспечив правильное согласование числа со словом "год", например: 20 — "двадцать лет", 32 — "тридцать два года", 41 — "сорок один год".
35. Дано целое число в диапазоне 100 — 999. Вывести строку — словесное описание данного числа, например: 256 — "двести пятьдесят шесть", 814 — "восемьсот четырнадцать".

36. В восточном календаре принят 60-летний цикл, состоящий из 12-летних подциклов, обозначаемых названиями цвета: зеленый, красный, желтый, белый и черный. В каждом подцикле годы носят названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. По номеру года вывести его название, если 1984 год был началом цикла — годом зеленой крысы.
37. Проверить истинность высказывания: "Квадратное уравнение  $A \cdot x^2 + B \cdot x + C = 0$  с данными коэффициентами  $A, B, C$  имеет вещественные корни".
38. Проверить истинность высказывания: "Данные числа  $x, y$  являются координатами точки, лежащей во второй координатной четверти".
39. Проверить истинность высказывания: "Данные числа  $x, y$  являются координатами точки, лежащей в первой или третьей координатной четверти".
40. Проверить истинность высказывания: "Точка с координатами  $(x, y)$  лежит внутри прямоугольника, левая верхняя вершина которого имеет координаты  $(x_1, y_1)$ , правая нижняя —  $(x_2, y_2)$ , а стороны параллельны координатным осям".
41. Проверить истинность высказывания: "Данное целое число является четным двузначным числом".
42. Проверить истинность высказывания: "Данное целое число является нечетным трехзначным числом".
43. Проверить истинность высказывания: "Среди трех данных целых чисел есть хотя бы одна пара совпадающих".
44. Проверить истинность высказывания: "Среди трех данных целых чисел есть хотя бы одна пара взаимно противоположных".
45. Проверить истинность высказывания: "Сумма цифр данного трехзначного числа является четным числом".
46. Проверить истинность высказывания: "Сумма двух первых цифр данного четырехзначного числа равна сумме двух его последних цифр".
47. Проверить истинность высказывания: "Данное четырехзначное число читается одинаково слева направо и справа налево".
48. Проверить истинность высказывания: "Все цифры данного трехзначного числа различны".
49. Проверить истинность высказывания: "Цифры данного трехзначного числа образуют возрастающую последовательность".
50. Проверить истинность высказывания: "Цифры данного трехзначного числа образуют возрастающую или убывающую последовательность".
51. Проверить истинность высказывания: "Цифры данного трехзначного числа образуют арифметическую прогрессию".
52. Проверить истинность высказывания: "Цифры данного трехзначного числа образуют геометрическую прогрессию".
53. Проверить истинность высказывания: "На плоскости точка  $A$  принадлежит отрезку с конечными точками  $B$  и  $C$ ".
54. Проверить истинность высказывания: "2 отрезка на плоскости, заданные парами целочисленных координат концевых точек пересекаются".
55. Даны координаты (как целые от 1 до 8) двух различных полей шахматной доски. Если ладья за один ход может перейти с одного поля на другое, вывести логическое значение True, в противном случае вывести значение False.
56. Даны координаты (как целые от 1 до 8) двух различных полей шахматной доски. Если король за один ход может перейти с одного поля на другое, вывести логическое значение True, в противном случае вывести значение False.

57. Даны координаты (как целые от 1 до 8) двух различных полей шахматной доски. Если слон за один ход может перейти с одного поля на другое, вывести логическое значение True, в противном случае вывести значение False.
58. Даны координаты (как целые от 1 до 8) двух различных полей шахматной доски. Если ферзь за один ход может перейти с одного поля на другое, вывести логическое значение True, в противном случае вывести значение False.
59. Даны координаты (как целые от 1 до 8) двух различных полей шахматной доски. Если конь за один ход может перейти с одного поля на другое, вывести логическое значение True, в противном случае вывести значение False.
60. Значения переменных X, Y, Z поменять местами так, чтобы они оказались упорядоченными по возрастанию.
61. Значения переменных X, Y, Z поменять местами так, чтобы они оказались упорядоченными по убыванию.
62. Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.
63. Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения.
64. Даны три переменные: X, Y, Z. Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.
65. Даны три переменные: X, Y, Z. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.
66. Даны целочисленные координаты точки на плоскости. Если точка не лежит на координатных осях, то вывести 0. Если точка совпадает с началом координат, то вывести 1. Если точка не совпадает с началом координат, но лежит на оси OX или OY, то вывести соответственно 2 или 3.
67. Даны вещественные координаты точки, не лежащей на координатных осях OX и OY. Вывести номер координатной четверти, в которой находится данная точка.
68. На числовой оси расположены три точки: A, B, C. Определить, какая из двух последних точек (B или C) расположена ближе к A, и вывести эту точку и ее расстояние от точки A.
69. Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Вывести порядковый номер этого числа.
70. Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.
71. Дан номер некоторого года (положительное целое число). Вывести число дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).
72. Дано целое число, лежащее в диапазоне от -999 до 999. Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.
73. Дано целое число, лежащее в диапазоне от 1 до 9999. Вывести строку — словесное описание данного числа вида "четное двузначное число", "нечетное четырехзначное число" и т.д.
74. На плоскости заданы две точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$ . Определить, какой из отрезков OA или OB образует больший угол с осью OX.
75. Отрезок на плоскости задается двумя не совпадающими концевыми точками  $X(x_1, x_2)$  и  $Y(y_1, y_2)$ . Из точки  $Z(z_1, z_2)$  к прямой, содержащей отрезок [X,Y], проводится перпендикуляр P. Определить, попадает ли перпендикуляр P на отрезок [X,Y] или на его продолжение.
76. Определить, пересекается ли прямая  $ax+by=u$  и отрезок с концами  $(x_1, y_1)$ ,  $(x_2, y_2)$ .

77. Треугольник на плоскости задается целочисленными координатами вершин. Для заданной точки  $Z(x,y)$  определить, принадлежит ли она стороне треугольника или лежит внутри или вне его.
78. Выяснить, пересекаются ли два отрезка на плоскости, и если да, то найти точку пересечения.
79. Можно ехать на такси со скоростью  $V1$  км/час и оплатой  $P1$  руб/км либо идти пешком со скоростью  $V2$  км/час бесплатно. Как с наименьшими затратами преодолеть путь  $S$  за время  $t$ , если это возможно? Каковы эти затраты?
80. Можно ли на прямоугольном участке застройки размером  $a$  на  $b$  метров разместить два дома размером в плане  $p$  на  $q$  и  $r$  на  $s$  метров? Дома располагаются параллельно сторонам участка.

## Циклы.

1. Введите с клавиатуры 10 целых чисел и определите их среднее арифметическое.
2. Введите целое число  $n$ . Посчитайте  $n!$   
 $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$ ;  $7! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7$
3. Напечатайте 20 первых степеней числа 2.
4. Разбейте заданное целое число на 2 слагаемых всеми различными способами. Разбиения, отличающиеся лишь порядком слагаемых, разными не считать.
5. Подсчитать количество цифр в десятичной записи целого неотрицательного числа.
6. Напечатать в убывающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.
7. Числа Фибоначчи определяются формулой  
 $f_0 = f_1 = 1$   
 $f_2 = f_1 + f_0$   
 $f_3 = f_2 + f_1$   
.....  
 $f_n = f_{n-1} + f_{n-2}$   
Определить  $f_{20}$  число Фибоначчи.
8. Дана последовательность из целых чисел, за которой следует 0. Определите номер наименьшего числа из этой последовательности.
9. Дана непустая последовательность из целых чисел, за которой следует 0. Определить сколько раз в этой последовательности меняется знак. (Например, в последовательности 1, -34, 8, 14, -5 знак меняется 3 раза.)
10. Напишите программу, которая складывает четные числа в промежутке от  $a$  до  $k$  включительно. Числа  $a, k$  задаются пользователем.
11. Напишите программу, которая вводит целое число, большее 1, и вычисляет сумму кубов от 1 до введенного числа. Например, если задано значение 3, то сумма кубов равна  $1 + 8 + 27 = 36$ .
12. Вычислить  $p = (1 - 1/2)(1 - 1/3)(1 - 1/4) \dots (1 - 1/n)$ . Для заданного целого числа  $n$ .
13. Посчитать количество всех введенных чисел, которые делятся на 5, но не делятся на 10. Концом ввода является число 0.
14. Написать программу, которая суммирует только трехзначные числа. Концом ввода считать 0.
15. Дана последовательность  $n$  целых чисел.  $n$  вводится пользователем. Посчитать сколько 0 и 1 в этой последовательности.
16. Дано  $n$  целых чисел,  $n$  вводится пользователем. Вывести каких чисел положительных или отрицательных больше в этой последовательности и на сколько.
17. Задано число  $m$ . Напечатать все числа от 1 до 1000, сумма цифр которых в десятичной записи равна заданному числу  $m$ .
18. Даны два целых числа  $A$  и  $B$  ( $A < B$ ). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания, а также количество  $N$  этих чисел.
19. Даны два целых числа  $A$  и  $B$  ( $A < B$ ). Вывести все целые числа, расположенные между данными числами (не включая сами эти числа), в порядке их убывания, а также количество  $N$  этих чисел.
20. Дано вещественное число  $A$  и целое число  $N$  ( $> 0$ ). Вывести  $A$  в степени  $N$ :  $A^N = A \cdot A \cdot \dots \cdot A$  (числа  $A$  перемножаются  $N$  раз).
21. Дано вещественное число  $A$  и целое число  $N$  ( $> 0$ ). Вывести все целые степени числа  $A$  от 1 до  $N$ .
22. Дано вещественное число  $A$  и целое число  $N$  ( $> 0$ ). Вывести  $1 + A + A^2 + A^3 + \dots + A^N$ .
23. Дано вещественное число  $A$  и целое число  $N$  ( $> 0$ ). Вывести  $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$ .
24. Дано целое число  $N$  ( $> 1$ ). Вывести наименьшее целое  $K$ , при котором выполняется неравенство  $3^K > N$ , и само значение  $3^K$ .
25. Дано целое число  $N$  ( $> 1$ ). Вывести наибольшее целое  $K$ , при котором выполняется неравенство  $3^K < N$ , и само значение  $3^K$ .
26. Дано вещественное число  $A$  ( $> 1$ ). Вывести наименьшее из целых чисел  $N$ , для которых сумма  $1 + 1/2 + \dots + 1/N$  будет больше  $A$ , и саму эту сумму.
27. Дано вещественное число  $A$  ( $> 1$ ). Вывести наибольшее из целых чисел  $N$ , для которых сумма  $1 + 1/2 + \dots + 1/N$  будет меньше  $A$ , и саму эту сумму.
28. Дано целое число  $N$  ( $> 0$ ). Вывести произведение  $1 \cdot 2 \cdot \dots \cdot N$ . Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.

29. Дано целое число  $N (> 0)$ . Если  $N$  — нечетное, то вывести произведение  $1 \cdot 3 \cdot \dots \cdot N$ ; если  $N$  — четное, то вывести произведение  $2 \cdot 4 \cdot \dots \cdot N$ . Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.
30. Дано целое число  $N (> 0)$ . Вывести сумму  $2 + 1/(2!) + 1/(3!) + \dots + 1/(N!)$  (выражение  $N!$  — " $N$  факториал" — обозначает произведение всех целых чисел от 1 до  $N$ :  $N! = 1 \cdot 2 \cdot \dots \cdot N$ ). Полученное число является приближенным значением константы  $e = \exp(1)$  ( $= 2.71828183\dots$ ).
31. Дано вещественное число  $X$  и целое число  $N (> 0)$ . Вывести  $1 + X + X^2/2! + \dots + X^N/N!$  ( $N! = 1 \cdot 2 \cdot \dots \cdot N$ ). Полученное число является приближенным значением функции  $\exp$  в точке  $X$ .
32. Дано вещественное число  $X$  и целое число  $N (> 0)$ . Вывести  $X - X^3/3! + X^5/5! - \dots + (-1)^N X^{2N+1}/(2N+1)!$  ( $N! = 1 \cdot 2 \cdot \dots \cdot N$ ). Полученное число является приближенным значением функции  $\sin$  в точке  $X$ .
33. Дано вещественное число  $X$  и целое число  $N (> 0)$ . Вывести  $1 - X^2/2! + X^4/4! - \dots + (-1)^N X^{2N}/(2N)!$  ( $N! = 1 \cdot 2 \cdot \dots \cdot N$ ). Полученное число является приближенным значением функции  $\cos$  в точке  $X$ .
34. Дано вещественное число  $X$  ( $|X| < 1$ ) и целое число  $N (> 0)$ . Вывести  $X - X^2/2 + X^3/3 - \dots + (-1)^{N-1} X^N/N$ . Полученное число является приближенным значением функции  $\ln$  в точке  $1+X$ .
35. Дано вещественное число  $X$  ( $|X| < 1$ ) и целое число  $N (> 0)$ . Вывести  $X - X^3/3 + X^5/5 - \dots + (-1)^N X^{2N+1}/(2N+1)$ . Полученное число является приближенным значением функции  $\arctg$  в точке  $X$ .
36. Дано целое число  $N (> 2)$  и две вещественные точки на числовой оси:  $A, B$  ( $A < B$ ). Отрезок  $[A, B]$  разбит на равные отрезки длины  $H$  с концами в  $N$  точках вида  $A, A + H, A + 2H, A + 3H, \dots, B$ . Вывести значение  $H$  и набор из  $N$  точек, образующий разбиение отрезка  $[A, B]$ .
37. Дано целое число  $N (> 2)$  и две вещественные точки на числовой оси:  $A, B$  ( $A < B$ ). Функция  $F(X)$  задана формулой  $F(X) = 1 - \sin(X)$ . Вывести значения функции  $F$  в  $N$  равноотстоящих точках, образующих разбиение отрезка  $[A, B]$ :  $F(A), F(A + H), F(A + 2H), \dots, F(B)$ .
38. Дано число  $D (> 0)$ . Последовательность чисел  $A_N$  определяется следующим образом:  $A_1 = 2, A_N = 2 + 1/A_{N-1}, N = 2, 3, \dots$ . Найти первый из номеров  $K$ , для которых выполняется условие  $|A_K - A_{K-1}| < D$ , и вывести этот номер, а также числа  $A_{K-1}$  и  $A_K$ .
39. Дано число  $D (> 0)$ . Последовательность чисел  $A_N$  определяется следующим образом:  $A_1 = 1, A_2 = 2, A_N = (A_{N-2} + A_{N-1})/2, N = 3, 4, \dots$ . Найти первый из номеров  $K$ , для которых выполняется условие  $|A_K - A_{K-1}| < D$ , и вывести этот номер, а также числа  $A_{K-1}$  и  $A_K$ .
40. Написать программу проверки пароля. Пусть пароль некоторое число фиксированное в программе. Программа печатает «введите пароль» и ждет ввода числа. Если введенное число совпадает выводится приветствие, если нет сообщение о неверном вводе и новая попытка. 0 – выход из программы.
41. Известен начальный вклад клиента в банк и процент годового дохода. Определить, через сколько лет вклад достигнет заданного размера.
42. Миша собирал орехи. В первый день он нашел 3 ореха, но, наловчившись, он каждый день стал собирать столько орехов, сколько собирал за все предыдущие дни. Сколько орехов собрал Миша за месяц?
43. Древнерусское умножение. Известен древнерусский прием умножения. Для умножения  $a$  и  $b$  поступаем так: делим  $a$  на 2, а  $b$  умножаем на 2, причем если делимое — число нечетное, то берем целую часть частного. С полученными числами проделываем то же самое, пока в частном не получится единица. Для получения произведения  $ab$  складываются только те числа правого столбика, слева от которого стоят нечетные числа.  
Пример:  $a = 42; b = 63$ :  

42	21	10	5	2	1
63	126	252	504	1008	2016

 $ab = 126 + 504 + 2016 = 2643$
44. Угадай число. Составить программу, по которой компьютер предлагает Вам задумать число в интервале от 1 до 100. Затем пытается угадать его наиболее быстрым способом. В ответ на сообщаемое компьютером число Вы отвечаете оно "равно", "больше" или "меньше" задуманного. В итоге сообщается задуманное число и количество сделанных попыток.
45. Спортлото. Составить программу, разыгрывающую тираж спортлото 6 из 36.
46. Пара кроликов каждый месяц дает приплод – двух кроликов (самку и самца), от которых через два месяца уже получается новый приплод. Сколько кроликов будет через год, если в начале была пара кроликов.
47. Натуральное число из  $n$  цифр является числом Армстронга, т.е. сумма его цифр возведенная в  $n$  степень, равна самому числу ( $153 = 1^3 + 5^3 + 3^3$ ). Получить все числа Армстронга для  $n=4$  и  $n=3$ .



48. Имеется  $n$  бактерий красного цвета. Через 1 такт времени красная бактерия меняется на зелёную, затем через 1 такт времени делится на красную и зелёную. Сколько будет всех бактерий через  $k$  тактов времени?
49. Дано число  $n$ . Выбросить из него все единицы и пятёрки, оставив порядок цифр  
пример: 527012 преобразуется в 2702
50. Перевести число из десятичной в двоичную систему счисления и из двоичной в десятичную систему счисления.

## Массивы.

1. Даны десять чисел. Вывести их среднее арифметическое.
2. Дано целое число  $N$  и набор из  $N$  вещественных чисел. Вывести сумму и произведение чисел из данного набора.
3. Дано целое число  $N$  и набор из  $N$  ненулевых целых чисел. Вывести в том же порядке все четные числа из данного набора и количество  $K$  таких чисел.
4. Дано целое число  $N$  и набор из  $N$  ненулевых целых чисел. Вывести в том же порядке номера всех нечетных чисел из данного набора и количество  $K$  таких чисел.
5. Даны целые числа  $K$ ,  $N$  и набор из  $N$  целых чисел. Если в наборе присутствует число, меньшее  $K$ , то вывести True; в противном случае вывести False.
6. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество элементов в наборе.
7. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести сумму всех положительных четных чисел из данного набора. Если требуемые числа в наборе отсутствуют, то вывести 0.
8. Дано целое число  $K$  и набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество чисел в наборе, меньших  $K$ .
9. Дано целое число  $K$  и набор ненулевых целых чисел; признак его завершения — число 0. Вывести номер первого числа в наборе, большего  $K$ . Если таких чисел в наборе нет, то вывести 0.
10. Дано целое число  $K$  и набор ненулевых целых чисел; признак его завершения — число 0. Вывести номер последнего числа в наборе, меньшего  $K$ . Если таких чисел в наборе нет, то вывести 0.
11. Дано целое число  $N$  и набор из  $N$  целых чисел. Вывести номера тех чисел в наборе, которые меньше своего левого соседа, и количество  $K$  таких чисел.
12. Дано целое число  $N$  и набор из  $N$  целых чисел. Вывести номера тех чисел в наборе, которые больше своего правого соседа, и количество  $K$  таких чисел.
13. Дано целое число  $N$  и набор из  $N$  вещественных чисел. Проверить, образует ли данный набор возрастающую последовательность. Если образует, то вывести True, если нет — вывести False.
14. Дано целое число  $N$  и набор из  $N$  вещественных чисел. Если данный набор образует убывающую последовательность, то вывести 0; в противном случае вывести номер первого числа, нарушающего закономерность.
15. Дано целое число  $N$  и набор из  $N$  целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между первыми двумя нулями (если первые нули идут подряд, то вывести 0).
16. Дано целое число  $N$  и набор из  $N$  целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между последними двумя нулями (если последние нули идут подряд, то вывести 0).
17. Даны целые числа  $K$ ,  $N$  и набор из  $N$  вещественных чисел:  $A_1, A_2, \dots, A_N$ . Вывести  $K$ -е степени чисел из данного набора:  $A_1^K, A_2^K, \dots, A_N^K$ .
18. Дано целое число  $N$  и набор из  $N$  вещественных чисел:  $A_1, A_2, \dots, A_N$ . Вывести следующие числа:  $A_1, A_2^2, \dots, A_{N-1}^{N-1}, A_N^N$ .
19. Дано целое число  $N$  и набор из  $N$  вещественных чисел:  $A_1, A_2, \dots, A_N$ . Вывести следующие числа:  $A_1^N, A_2^{N-1}, \dots, A_{N-1}^2, A_N$ .
20. Дана последовательность из 10 целых чисел. Найти величину наибольшего среди отрицательных чисел этой последовательности.
21. Дана последовательность из 10 целых чисел. Вычислить разность между максимальным и минимальным числом этой последовательности.
22. Дано 20 действительных чисел. Определить образуют ли они возрастающую последовательность.
23. Дана последовательность из 20 целых чисел. Определить со сколько отрицательных чисел она начинается.
24. Дано 20 действительных чисел. Определить сколько из них принимают максимальное значение.

25. Дано 20 действительных чисел. Определить сколько из них принимают минимальное значение.
26. Дано 20 целых чисел. Определить, сколько из них больше своих «соседей», т.е. предыдущего и последующего.
27. Даны 20 целых чисел. Получить удвоенную сумму всех положительных членов этой последовательности.
28. Найти минимальный и максимальный из данных десяти элементов.
29. Найти номера минимального<sub>1</sub>|максимального<sub>2</sub> из данных десяти элементов.
30. Найти номера первого<sub>1</sub>|последнего<sub>2</sub> минимального<sub>3</sub>|максимального<sub>4</sub> из данных десяти целочисленных элементов.
31. Найти количество минимальных<sub>1</sub>|максимальных<sub>2</sub> из данных десяти целочисленных элементов.
32. Найти минимальный<sub>1</sub>|максимальный<sub>2</sub> четный<sub>3</sub>|нечетный<sub>4</sub> из данных десяти ненулевых целочисленных элементов. Если требуемые элементы отсутствуют, то вывести 0.
33. Найти [минимальный положительный]<sub>1</sub>|[максимальный отрицательный]<sub>2</sub> из данных десяти элементов. Если требуемые элементы отсутствуют, то вывести 0.
34. Даны числа  $a, b$  ( $0 < a < b$ ) и набор из десяти элементов. Найти минимальный<sub>1</sub>|максимальный<sub>2</sub> из элементов, содержащихся в интервале  $(a, b)$ . Если требуемые элементы отсутствуют, то вывести  $-1$ .
35. Дан набор из десяти целочисленных элементов. Найти количество элементов, расположенных перед<sub>1</sub>|после<sub>2</sub> первого<sub>3</sub>|последнего<sub>4</sub> минимального<sub>5</sub>|максимального<sub>6</sub>.
36. Найти номер первого<sub>1</sub>|последнего<sub>2</sub> экстремального (то есть минимального или максимального) из данных десяти целочисленных элементов.
37. Дан набор из десяти целочисленных элементов. Найти количество элементов, содержащихся между первым и последним минимальным<sub>1</sub>|максимальным<sub>2</sub>. Если в наборе имеется единственный минимальный<sub>1</sub>|максимальный<sub>2</sub> элемент, то вывести 0.
38. Найти два наименьших<sub>1</sub>|наибольших<sub>2</sub> из данных десяти элементов.
39. Дан набор из десяти целочисленных элементов. Найти максимальное количество подряд идущих минимальных<sub>1</sub>|максимальных<sub>2</sub> элементов.
40. Дан массив размера  $N$ . Вывести его элементы в обратном порядке.
41. Дан массив размера  $N$ . Вывести вначале его элементы с четными<sub>1</sub>|нечетными<sub>2</sub> индексами, а затем — с нечетными<sub>1</sub>|четными<sub>2</sub>.
42. Дан целочисленный массив  $A$  размера 10. Вывести номер первого<sub>1</sub>|последнего<sub>2</sub> из тех его элементов  $A[i]$ , которые удовлетворяют двойному неравенству:  $A[1] < A[i] < A[10]$ . Если таких элементов нет, то вывести 0.
43. Дан целочисленный массив размера  $N$ . Преобразовать его, прибавив к четным<sub>1</sub>|нечетным<sub>2</sub> числам первый<sub>3</sub>|последний<sub>4</sub> элемент. Первый и последний элементы массива не изменять.
44. Дан целочисленный массив размера  $N$ . Вывести вначале все его четные<sub>1</sub>|нечетные<sub>2</sub> элементы, а затем — нечетные<sub>1</sub>|четные<sub>2</sub>.
45. Поменять местами минимальный и максимальный элементы массива размера 10.
46. Заменить все положительные<sub>1</sub>|отрицательные<sub>2</sub> элементы целочисленного массива размера 10 на значение минимального<sub>3</sub>|максимального<sub>4</sub>.
47. Дан массив размера 10. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.
48. Дан массив размера  $N$ . Осуществить циклический сдвиг элементов массива влево<sub>1</sub>|вправо<sub>2</sub> на одну позицию.
49. Дан массив размера  $N$  и число  $k$  ( $0 < k < 5, k < N$ ). Осуществить циклический сдвиг элементов массива влево<sub>1</sub>|вправо<sub>2</sub> на  $k$  позиций.
50. Проверить, образуют ли элементы целочисленного массива размера  $N$  арифметическую<sub>1</sub>|геометрическую<sub>2</sub> прогрессию. Если да, то вывести разность<sub>1</sub>|знаменатель<sub>2</sub> прогрессии, если нет — вывести 0.
51. Дан массив ненулевых целых чисел размера  $N$ . Проверить, чередуются ли в нем [четные и нечетные]<sub>1</sub>|[положительные и отрицательные]<sub>2</sub> числа. Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.

52. Дан массив размера  $N$ . Найти количество его локальных минимумов<sub>1</sub>|максимумов<sub>2</sub>.
53. Дан массив размера  $N$ . Найти максимальный<sub>1</sub>|минимальный<sub>2</sub> из его локальных минимумов<sub>1</sub>|максимумов<sub>2</sub>.
54. Дан массив размера  $N$ . Определить количество участков, на которых его элементы монотонно возрастают<sub>1</sub>|убывают<sub>2</sub>.
55. Дан массив размера  $N$ . Определить количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).
56. Дано вещественное число  $R$  и массив размера  $N$ . Найти элемент массива, который наиболее<sub>1</sub>|наименее<sub>2</sub> близок к данному числу.
57. Дано вещественное число  $R$  и массив размера  $N$ . Найти два элемента массива, сумма которых наиболее<sub>1</sub>|наименее<sub>2</sub> близка к данному числу.
58. Дан массив размера  $N$ . Найти номера двух ближайших чисел из этого массива.
59. Дан целочисленный массив размера  $N$ . Определить максимальное количество его одинаковых элементов.
60. Дан целочисленный массив размера  $N$ . Удалить из массива все элементы, встречающиеся [менее двух раз]<sub>1</sub>[[более двух раз]<sub>2</sub>[[ровно два раза]<sub>3</sub>[[ровно три раза]<sub>4</sub>.
61. Дан целочисленный массив размера  $N$ . Если он является перестановкой, то есть содержит все числа от 1 до  $N$ , то вывести 0, в противном случае вывести номер первого недопустимого элемента.
62. Дан массив размера  $N$ . Преобразовать его, вставив перед<sub>1</sub>|после<sub>2</sub> каждого положительного<sub>3</sub>|отрицательного<sub>4</sub> элемента нулевой элемент.
63. Дан целочисленный массив размера  $N$ . Назовем *серией* группу подряд идущих одинаковых элементов, а длиной серии — количество этих элементов (длина серии может быть равна 1). Вывести массив, содержащий длины всех серий исходного массива.
64. Дан целочисленный массив размера  $N$ . Преобразовать массив, увеличив<sub>1</sub>|уменьшив<sub>2</sub> каждую его серию на один элемент.
65. Дан целочисленный массив размера  $N$ . Преобразовать массив, увеличив первую<sub>1</sub>|последнюю<sub>2</sub>|все<sub>3</sub> серии наибольшей длины на один элемент.
66. Дан целочисленный массив размера  $N$ . Вставить перед<sub>1</sub>|после<sub>2</sub> каждой серии нулевой элемент.
67. Дано число  $k$  и целочисленный массив размера  $N$ . Поменять местами первую<sub>1</sub>|последнюю<sub>2</sub> и  $k$ -ю серии массива. Если серий в массиве меньше  $k$ , то вывести массив без изменений.
68. Дано число  $k$  и целочисленный массив размера  $N$ . Удалить из массива все серии, длина которых меньше<sub>1</sub>|равна<sub>2</sub>|больше<sub>3</sub>  $k$ .
69. Дано число  $k$  и целочисленный массив размера  $N$ . Заменить каждую серию, длина которой меньше<sub>1</sub>|равна<sub>2</sub>|больше<sub>3</sub>  $k$ , на один нулевой элемент.
70. Даны два массива  $A$  и  $B$  размера 5, элементы которых упорядочены по возрастанию<sub>1</sub>|убыванию<sub>2</sub>. Объединить эти массивы так, чтобы результирующий массив остался упорядоченным.
71. Упорядочить массив размера  $N$  по возрастанию<sub>1</sub>|убыванию<sub>2</sub>.
72. Дан массив размера  $N$ . Вывести индексы массива в том порядке, в котором соответствующие им элементы образуют возрастающую<sub>1</sub>|убывающую<sub>2</sub> последовательность.
73. Дана точка  $A$  и множество  $B$  из  $N$  точек. Найти номер точки из множества  $B$ , наиболее близкой<sub>1</sub>|удаленной<sub>2</sub> от точки  $A$ .
74. Дано множество  $A$  из  $N$  точек. Среди всех точек этого множества, лежащих в первой<sub>1</sub>|второй<sub>2</sub>|третьей<sub>3</sub>|четвертой<sub>4</sub> четверти, найти точку, наиболее близкую<sub>5</sub>|удаленную<sub>6</sub> от начала координат. Если таких точек нет, то вывести точку с нулевыми координатами.
75. Дано множество  $A$  из  $N$  точек. Найти пару различных точек этого множества с минимальным<sub>1</sub>|максимальным<sub>2</sub> расстоянием между ними и само это расстояние (точки выводятся в том же порядке, в котором они перечислены при задании множества  $A$ ).
76. Дано множество  $A$  из  $N$  точек. Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна<sub>1</sub>|максимальна<sub>2</sub>, и саму эту сумму.

77. Даны множества  $A$  и  $B$ , состоящие соответственно из  $N_1$  и  $N_2$  точек. Найти минимальное<sub>1</sub>|максимальное<sub>2</sub> расстояние между точками этих множеств и сами точки, расположенные на этом расстоянии.
78. Дано множество  $A$  из  $N$  точек. Найти наименьший<sub>1</sub>|наибольший<sub>2</sub> периметр треугольника, вершины которого принадлежат различным точкам множества  $A$ , и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества  $A$ ).
79. Дано множество  $A$  из  $N$  точек с целочисленными координатами. Порядок на координатной плоскости определим следующим образом:  $(x_1, y_1) < (x_2, y_2)$ , если либо  $x_1 < x_2$ , либо  $x_1 = x_2$  и  $y_1 < y_2$ . Расположить точки данного множества по возрастанию<sub>1</sub>|убыванию<sub>2</sub> в соответствии с указанным порядком.
80. Циклически сдвинуть элементы таблицы из  $N$  элементов на  $k$  мест влево за время  $O(N)$ . Составить программу, аргументами которой являются длина  $N$  линейной таблицы  $A$  и сама таблица  $A[1:N]$ . Использовать дополнительные таблицы нельзя.

## Строки.

1. Задана символьная строка. Поменять попарно местами символы, расположенные на четных и нечетных позициях.
2. Задана символьная строка, содержащая буквы и цифры. Посчитать произведение цифр строки.
3. Задана символьная строка, содержащая две точки. Определить и напечатать количество символов между точками.
4. Задана символьная строка, состоящая из букв и чисел. Напечатать сумму чисел.
5. Удалить из символьной строки все слоги «ма».
6. В символьной строке удалить все знаки препинания.
7. Задана символьная строка. Определить сколько разных символов она содержит.
8. Заданы две символьные строки. Все ли буквы первого слова встречаются во втором.
9. Заменить в заданной строке все буквы «о» на букву «а».
10. В заданной строке заменить все буквы «б» на сочетание «ку».
11. Сколько раз заданное слово встречается в заданной строке.
12. Заданы два слова. Определить, не является ли одно обращением (перевертышем) другого.
13. Отредактировать предложение, удаляя все слова на четных позициях.
14. Дана строка, содержащая цифры, увеличить все цифры в этой строке вдвое.
15. В заданном предложении найти все слова, начинающиеся с заданной приставки.
16. Перед каждым словом предложения добавить сочетание «по».
17. Определить, является ли заданное слово симметричным (симметричное слово - кабак)
18. Определить, является ли заданное шестизначное число счастливым.
19. Шифр Цезаря. Алфавит представляет цепочку пронумерованных букв. Каждая буква текста заменяется на букву, находящуюся на 4 символа левее от шифруемой. При выходе за нумерацию двигаемся по кругу).
20. Задана строка, определить частоту встречаемости каждой буквы в тексте.
21. Задана строка, определить частоту встречаемости каждого слова в тексте.
22. Удалить из заданной строки-предложения лишние пробелы.
23. Шифр гаммирования. Над текстом записывается (если необходимо с повторением) ключевое слово, затем складываем номера букв в алфавите, получаем шифрованное сообщение.
24. Задана строка-предложение. Напечатать самое длинное слово строки. Слова в предложении разделены одним или несколькими пробелами.
25. Задана строка-предложение. Расположить слова в порядке убывания их длины в предложении. Слова в предложении разделены одним или несколькими пробелами.
26. Задано строка-предложение. Расположить слова в алфавитном порядке. Слова в предложении разделены одним или несколькими пробелами.
27. Дан текст, определить кол-во слов "кот". Слова в предложении разделены одним или несколькими пробелами.
28. Получить из слова a, вычеркиванием некоторого кол-ва букв, слово b.
29. Дано предложение. Сколько слов является перевёртышами. Слова в предложении разделены одним или несколькими пробелами.
30. Дано предложение заканчивающееся '!', '!', '?'. Разделитель слов - пробел. В скольких словах предложения имеется словосочетание "ка".
31. Слова в предложении разделены одним или несколькими пробелами. Определить слово с максимальным числом букв "а" и количество таких букв "а".

32. Дано предложение. Найти в каком из слов, больше четырёх символов, буква "а" встречается реже. Слова в предложении разделены одним или несколькими пробелами.
33. Дан текст. Удалить в нём все слова "функция". Слова в предложении разделены одним или несколькими пробелами.
34. Заменить любой пунктуационный символ ( , : ? и др.) в строке многоточием.
35. Дан текст. Определить все слова оканчивающиеся на "ая". Слова в предложении разделены одним или несколькими пробелами.
36. Дан текст. Сколько в нём слов "что". Слова в предложении разделены одним или несколькими пробелами.
37. Дана строка текста, состоящая из слов разделенных одним из знаков [#,\$,\*,-]. Если кол-во слов в предложении чётно, поменяйте местами два центральных слова, а если нечётно удалите одно центральное слово.
38. Задаётся два слова. Выяснить являются ли они анаграммами (слова, составленные из одних и тех же букв). Слова в предложении разделены одним или несколькими пробелами.
39. Дано число n. Выбросить из него все единицы и пятёрки, оставив порядок цифр.  
пример: 527012 преобразуется в 2702
40. Перевести число из десятичной в двоичную систему счисления.
41. Перевести число из двоичной в десятичную систему счисления.
42. Дан текст. Отбросить повторяющиеся слова. Слова в предложении разделены одним или несколькими пробелами.
43. Дан текст. Вывести повторяющиеся слова и их кол-во. Слова в предложении разделены одним или несколькими пробелами.
44. Напечатать словарь состоящий из четырёх букв неповторяющихся в слове.
45. Определить можно ли из символов входящих в строку a, составить строку c.
46. Дано предложение t. Заменить в нем слово 'потоп' словом 'потопкот'. Слова в предложении разделены одним или несколькими пробелами.
47. Дан текст. Слова в предложении разделены одним или несколькими пробелами и знаками пунктуации. Определить все ли слова в начале предложения написаны с большой буквы.
48. Проверить заданный текст на ошибки написания букв Ы Ю Я после шипящих Ж Ч Ш Щ. Учесть исключения (ЖЮРИ, БРОШЮРА, ПАРАШЮТ)
49. В словах текста добавить знаки переноса, используя правило: в каждом слоге больше 1 символа и хотя бы одна гласная.
50. Дан текст. Найти самое длинное слово без повторяющихся букв. Слова в предложении разделены одним или несколькими пробелами.
51. Задано выражение, содержащее вложенные скобки (не более 3 вложений). Заменить внешние на { }, следующее вложение на [ ], а внутренние оставить без изменения.
52. Задано выражение, состоящее из переменных (одна буква), цифр и арифметических действий. Правильно ли записано выражение? (последовательность операнд операция операнд).
53. В заданном тексте заменить все буквы «а» на «о», а «о» на «а».
54. Задан текст из нескольких предложений. Определить хотя бы одно слово встречающееся в каждом предложении.
55. Задан набор слов. Найти однокоренные слова.
56. Задано выражение из чисел и арифметических действий + и -. Подсчитать значение выражения.
57. Задано выражение из чисел и арифметических действий \* и /. Подсчитать значение выражения.
58. Задано предложение. Слова разделены одним или несколькими пробелами. Составить новое предложение из «перевернутых» слов исходного.
59. Не с глаголами пишется раздельно. Глаголы оканчиваются чаще всего на: «тся», «ться», «ал», «ял», «ил», «ел». Проверить правильность написания нес глаголами в предложении.

## Матрицы

1. Дано натуральное число  $n$ , действительная матрица размера  $n \times m$ . Найти среднее арифметическое каждого из столбцов.
2. Дано натуральное число  $n$ , действительная матрица размера  $n \times m$ . Найти среднее арифметическое каждого из столбцов, имеющих четные номера.
3. Дано натуральное число  $n$ . Выяснить, сколько положительных элементов содержит матрица  $[a_{ij}]_{i,j=1,\dots,n}$  если  $a_{ij} = \sin(i+j/2)$ .
4. Дано натуральное число  $n$ . Выяснить, сколько положительных элементов содержит матрица  $[a_{ij}]_{i,j=1,\dots,n}$  если  $a_{ij} = \cos(i^2+n)$ .
5. Дано натуральное число  $n$ . Выяснить, сколько положительных элементов содержит матрица  $[a_{ij}]_{i,j=1,\dots,n}$  если  $a_{ij} = \sin((i^2-j^2)/n)$ .
6. Дана действительная матрица размера  $m \times n$  в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.
7. Дано натуральное число  $n$ , целые числа  $a_1, \dots, a_m$  и целочисленная квадратная матрица порядка  $m$ . Строку с номером  $i$  матрицы назовем отмеченной, если  $a_i > 0$ , и неотмеченной в противном случае. Нужно все элементы, расположенные в отмеченных строках матрицы, преобразовать по правилу: отрицательные элементы заменить на  $-1$ , положительные - на  $1$ , нулевые оставить без изменения.
8. Дано натуральное число  $n$ , целые числа  $a_1, \dots, a_m$  и целочисленная квадратная матрица порядка  $m$ . Строку с номером  $i$  матрицы назовем отмеченной, если  $a_i > 0$ , и неотмеченной в противном случае. Подсчитать число отрицательных элементов матрицы, расположенных в отмеченных строках.
9. Дана действительная квадратная матрица порядка  $n$ . Заменить нулями все ее элементы, расположенные на главной диагонали и выше нее.
10. Даны действительные числа  $x_1, x_2, \dots, x_n$ . Получить действительную квадратную матрицу порядка  $n$ :
$$\begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots \\ x_1^n & x_2^n & \dots & x_n^n \end{bmatrix}$$
11. Даны действительные числа  $x_1, x_2, \dots, x_n$ . Получить действительную квадратную матрицу порядка  $n$ :
$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \dots & \dots & \dots & \dots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{bmatrix}$$
12. Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$  равные суммам элементов строк.
13. Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$  равные произведениям элементов строк.
14. Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$  равные наименьшим значениям элементов строк.
15. Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$  равные значениям средних арифметических элементов строк.
16. Дана действительная матрица размера  $m \times n$ . Определить числа  $b_1, \dots, b_m$  равные разностям наибольших и наименьших значений элементов строк.
17. Дано натуральное число  $n$ , действительная матрица  $[a_{ij}]_{i=1,\dots,n; j=1,\dots,n}$ . Получить последовательность элементов главной диагонали  $a_{11}, a_{22}, \dots, a_{nn}$ .
18. Все элементы с наибольшим значением в данной целочисленной квадратной матрице порядка  $n$  заменить нулями.
19. Дана действительная матрица размера  $n \times m$ . Найти среднее арифметическое наибольшего и наименьше-



го значений ее элементов.

20. Дана действительная матрица размера  $n \times m$ . Найти значение наибольшего по модулю элемента матрицы, а также индексы какого-нибудь элемента с найденным значением модуля.
21. Дана действительная матрица размера  $n \times m$ . Найти сумму наибольших значений элементов ее строк.
22. В данной действительной квадратной матрице порядка  $n$  найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.
23. В данной действительной матрице размера  $n \times m$  поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением. Предполагается, что эти элементы единственны.
24. В данной квадратной целочисленной матрице порядка  $n$  указать индексы всех элементов с наибольшим значением.
25. Дана действительная матрица размера  $n \times m$ , все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.
26. Дана действительная матрица размера  $n \times m$ . Получить последовательность  $b_1, \dots, b_n$  где  $b_k$  — это наибольшее из значений элементов  $k$ -й строки.
27. Дана действительная матрица размера  $n \times m$ . Получить последовательность  $b_1, \dots, b_n$  где  $b_k$  — это сумма наибольшего и наименьшего из значений элементов  $k$ -й строки.
28. Дана действительная матрица размера  $n \times m$ . Получить последовательность  $b_1, \dots, b_n$  где  $b_k$  — это число отрицательных элементов в  $k$ -й строке.
29. Дана действительная матрица размера  $n \times m$ . Получить последовательность  $b_1, \dots, b_n$  где  $b_k$  — это произведение квадратов тех элементов  $k$ -й строки, модули которых принадлежат отрезку  $[1, 1.5]$ .
30. Даны натуральное число  $n$ , целочисленная матрица  $[a_{ij}]_{i=1,2; j=1,\dots,m}$ . Найти сумму тех из элементов  $a_{2j}(j=1,\dots,m)$ , для которых  $a_{1j}$  имеет значение наибольшего среди значений  $a_{11}, a_{12}, \dots, a_{1m}$ .
31. Дана целочисленная квадратная матрица порядка  $n$ . Найти наименьшее из значений элементов столбца, который обладает наибольшей суммой модулей элементов. Если таких столбцов несколько, то взять первый из них.
32. Дано натуральное число  $n$ , целочисленная квадратная матрица порядка  $n$ . Получить  $b_1, \dots, b_n$  где  $b_k$  — это наименьшее из значений элементов, находящихся в начале  $i$ -й строки матрицы до элемента, принадлежащего главной диагонали, включительно.
33. Дано натуральное число  $n$ , целочисленная квадратная матрица порядка  $n$ . Получить  $b_1, \dots, b_n$  где  $b_k$  — это значение первого по порядку положительного элемента  $i$ -й строки (если таких элементов нет, то принять  $b_i = -1$ ).
34. Дано натуральное число  $n$ , целочисленная квадратная матрица порядка  $n$ . Получить  $b_1, \dots, b_n$  где  $b_k$  — это сумма элементов, расположенных за первым отрицательным элементом в  $i$ -й строке (если все элементы строки неотрицательны, то принять  $b_i = 100$ ).
35. Дано натуральное число  $n$ , целочисленная квадратная матрица порядка  $n$ . Получить  $b_1, \dots, b_n$  где  $b_k$  — это сумма элементов, предшествующих последнему отрицательному элементу  $i$ -й строки (если все элементы строки неотрицательны, то принять  $b_i = -1$ ).
36. Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк все элементы которых - нули.
37. Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк элементы в каждой из которых одинаковы.
38. Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк все элементы которых четны.
39. Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк элементы каждой из которых образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую).
40. Дана целочисленная квадратная матрица порядка  $n$ . Найти номера строк элементы которых образуют симметричные последовательности (палиндромы).
41. Дано натуральное число  $n$ , действительное число  $x$ , действительная матрица размера  $n \times 2n$ . Получить последовательность  $b_1, \dots, b_n$  из нулей и единиц, где  $b_i = 1$ , если элементы  $i$ -й строки матрицы не превос-

ходят  $x_i$  и  $b_i = 0$  в противном случае.

42. Дана действительная квадратная матрица порядка  $n$ . Построить последовательность действительных чисел  $a_1, \dots, a_n$  по правилу: если в  $i$ -й строке матрицы элемент, принадлежащий главной диагонали, отрицателен то  $a_i$  равно сумме элементов  $i$ -й строки, предшествующих первому отрицательному элементу; в противном случае  $a_i$  равно сумме последних элементов  $i$ -й строки, начиная с первого по порядку неотрицательного элемента.
43. Дана действительная квадратная матрица порядка  $n$ . В строках с отрицательным элементом на главной диагонали найти сумму всех элементов.
44. Дана действительная квадратная матрица порядка  $n$ . В строках с отрицательным элементом на главной диагонали найти наибольший из всех элементов.
45. Дана действительная квадратная матрица порядка  $n$ . Рассмотрим те элементы, которые расположены в строках, начинающихся с отрицательного элемента. Найти суммы тех из них, которые расположены соответственно ниже, выше и на главной диагонали.
46. Дана действительная квадратная матрица порядка  $n$ . Получить целочисленную квадратную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше элемента, расположенного в его строке на главной диагонали, и равен нулю в противном случае.
47. Дана действительная квадратная матрица порядка  $n$ . Получить  $x_1 * x_n + x_2 * x_{n-1} + \dots + x_n * x_1$  где  $x_k$  наибольшее значение элементов  $k$ -й строки данной матрицы.
48. Даны действительная квадратная матрица порядка  $n$ , натуральные числа  $i, j$  ( $1 \leq i \leq n, 1 \leq j \leq n$ ). Из матрицы удалить  $i$ -ю строку и  $j$ -й столбец.
49. Даны натуральное число  $n \geq 2$ , действительная квадратная матрица порядка  $n$ . Построить последовательность  $b_1, \dots, b_n$  из нулей и единиц, в которой  $b_i = 1$  тогда и только тогда, когда элементы  $i$ -й строки матрицы образуют возрастающую последовательность.
50. Даны натуральное число  $n \geq 2$ , действительная квадратная матрица порядка  $n$ . Построить последовательность  $b_1, \dots, b_n$  из нулей и единиц, в которой  $b_i = 1$  тогда и только тогда, когда элементы  $i$ -й строки матрицы образуют возрастающую или убывающую последовательность.
51. Дана целочисленная квадратная матрица порядка 15. Выяснить, имеются ли в матрице ненулевые элементы, и если имеются, то указать индексы всех ненулевых элементов.
52. Даны натуральные числа  $i, j$ , действительная матрица размера  $m * n$  ( $1 \leq i < j \leq n$ ). Поменять в матрице местами  $i$ -й и  $j$ -й столбцы.
53. Даны натуральное число  $n$ , действительная квадратная матрица порядка  $n$ . Построить последовательность  $b_1, \dots, b_n$  из нулей и единиц, в которой  $b_i = 1$  тогда и только тогда, когда в  $i$ -й строке матрицы есть хотя бы один отрицательный элемент.
54. С помощью  $[x_{ij}]_{i=1,2; j=1, \dots, n}$  - действительной матрицы на плоскости задано  $n$  точек так, что  $x_{1j}, x_{2j}$  - координаты  $j$ -й точки. Точки попарно соединены отрезками. Найти длину наибольшего отрезка.
55. Даны натуральные числа  $n$  и  $m$ , действительное число  $r$ , действительная матрица размера  $n * m$ . Получить значение  $b_1 r^{n-1} + b_2 r^{n-2} + \dots + b_n$ , где  $b_k$  - первый по порядку положительный элемент в  $k$ -й строке матрицы ( $k=1, \dots, n$ ); если в  $k$ -й строке нет положительных элементов, то  $b_k = 0.5$ .
56. Найти сумму квадратов тех элементов  $a_{ij}$  матрицы  $[a_{ij}]_{i,j=1, \dots, n}$ , для которых выполнено  $2 \leq i \leq n, 2 \leq j \leq n, a_{ij} \geq (a_{i-1,j} + a_{i,j-1} + a_{i+1,j} + a_{i,j+1})/4$
57. Дана действительная квадратная матрица порядка  $n$ . Вычислить сумму тех из ее элементов, расположенных на главной диагонали и выше нее, которые превосходят по величине все элементы, расположенные ниже главной диагонали. Если на главной диагонали и выше нее нет элементов с указанным свойством, то ответом должно служить сообщение об этом.
58. Дана целочисленная матрица  $[a_{ij}]_{i,j=1, \dots, n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  - это  $\sum_{j=1}^n a_{kj}^2$ .
59. Дана целочисленная матрица  $[a_{ij}]_{i,j=1, \dots, n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  - это  $\sum_{j=1}^n (-1)^{i+j} a_{ij}$ .

60. Дана целочисленная матрица  $[a_{ij}]_{i,j=1,\dots,n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  – это  $\prod_{j=1}^n a_{kj}$ .
61. Дана целочисленная матрица  $[a_{ij}]_{i,j=1,\dots,n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  – это  $\sum_{j=1}^n |a_{kj}|$ .
62. Дана целочисленная матрица  $[a_{ij}]_{i,j=1,\dots,n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  – это  $\prod_j a_{ji}$ , для всех таких  $j$ , что  $1 < a_{ji} \leq n$ .
63. Дана целочисленная матрица  $[a_{ij}]_{i,j=1,\dots,n}$ . Получить  $b_1, \dots, b_n$ , где  $b_k$  – это  $\max_{1 \leq j \leq n} a_{kj} * \min_{1 \leq j \leq n} a_{ji}$ .
64. Будем называть соседями элемента с индексами  $i, j$  некоторой матрицы такие элементы этой матрицы, соответствующие индексы которых отличаются от  $i$  и  $j$  не более чем на единицу. Для данной целочисленной матрицы  $[a_{ij}]_{i=1,\dots,n; j=1,\dots,m}$  найти матрицу из нулей и единиц  $[b_{ij}]_{i=1,\dots,n; j=1,\dots,m}$ , элемент которой  $b_{ij}$  равен единице, когда все соседи  $a_{ij}$  меньше самого  $a_{ij}$ .
65. Будем называть соседями элемента с индексами  $i, j$  некоторой матрицы такие элементы этой матрицы, соответствующие индексы которых отличаются от  $i$  и  $j$  не более чем на единицу. Для данной целочисленной матрицы  $[a_{ij}]_{i=1,\dots,n; j=1,\dots,m}$  найти матрицу из нулей и единиц  $[b_{ij}]_{i=1,\dots,n; j=1,\dots,m}$ , элемент которой  $b_{ij}$  равен единице, когда все соседи  $a_{ij}$  и само  $a_{ij}$  равны нулю.
66. Будем называть соседями элемента с индексами  $i, j$  некоторой матрицы такие элементы этой матрицы, соответствующие индексы которых отличаются от  $i$  и  $j$  не более чем на единицу. Для данной целочисленной матрицы  $[a_{ij}]_{i=1,\dots,n; j=1,\dots,m}$  найти матрицу из нулей и единиц  $[b_{ij}]_{i=1,\dots,n; j=1,\dots,m}$ , элемент которой  $b_{ij}$  равен единице, когда среди соседей  $a_{ij}$  есть не менее двух совпадающих с  $a_{ij}$ .
67. Даны две целочисленные квадратные матрицы порядка  $n$ . Найти последовательность из нулей и единиц  $b_1, \dots, b_n$  такую, что  $b_i = 1$ , когда все элементы  $i$ -й. строки первой матрицы для соответствующих элементов  $i$ -й строки второй матрицы имеют одинаковые знаки.
68. Даны две целочисленные квадратные матрицы порядка  $n$ . Найти последовательность из нулей и единиц  $b_1, \dots, b_n$  такую, что  $b_i = 1$ , когда все элементы  $i$ -й. строки первой матрицы больше соответствующих элементов  $i$ -й строки второй матрицы.
69. Даны две целочисленные квадратные матрицы порядка  $n$ . Найти последовательность из нулей и единиц  $b_1, \dots, b_n$  такую, что  $b_i = 1$ , когда все элементы  $i$ -х строк первой и второй матриц отрицательны.
70. Даны две целочисленные квадратные матрицы порядка  $n$ . Найти последовательность из нулей и единиц  $b_1, \dots, b_n$  такую, что  $b_i = 1$ , когда  $i$ -е строки первой и второй матриц содержат вместе не более трех положительных элементов.
71. Даны две целочисленные квадратные матрицы порядка  $n$ . Найти последовательность из нулей и единиц  $b_1, \dots, b_n$  такую, что  $b_i = 1$ , когда количество отрицательных и неотрицательных элементов  $i$ -й строки первой матрицы совпадает соответственно с количеством отрицательных и неотрицательных элементов  $i$ -й строки второй матрицы.
72. Таблица футбольного чемпионата задана квадратной матрицей порядка  $n$ , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2 - выигрыш, 1 - ничья, 0 - проигрыш). Найти число команд, имеющих больше побед, чем поражений.
73. Таблица футбольного чемпионата задана квадратной матрицей порядка  $n$ , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2 - выигрыш, 1 - ничья, 0 - проигрыш). Определить номера команд, прошедших чемпионат без поражений.
74. Таблица футбольного чемпионата задана квадратной матрицей порядка  $n$ , в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2 - выигрыш, 1 - ничья, 0 - проигрыш). Выяснить, имеется ли хотя бы одна команда, выигравшая более половины игр.
75. Даны натуральные числа  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Числа  $x_i, y_i$  рассматриваются как координаты  $i$ -й точки ( $i=1, \dots, n$ ). Обозначим через  $g_{ij}$  расстояние от  $i$ -й точки до  $j$ -й. Получить на экране заданные точки и со-

единить отрезком  $i$ -ю точку с  $j$ -й в том случае, если выполняется по крайней мере одно условие:

- $r_{ij}$  имеет наибольшее значение из  $r_{i1}, r_{i2}, r_{i3}, \dots, r_{in}$ ;
- $r_{ji}$  имеет наибольшее значение из  $r_{j1}, r_{j2}, r_{j3}, \dots, r_{jn}$ .

76. Дана целочисленная квадратная матрица порядком. Каждый элемент матрицы ставится в соответствие точке, принадлежащей квадратной области экрана размером  $n \times n$  точек. Левый верхний угол области имеет координаты 0,0. Соответствие между элементами матрицы и точками области экрана устанавливается следующим образом: элемент матрицы, стоящий в строке с номером  $i$  и в столбце с номером  $j$ , соответствует точке экрана, находящейся на пересечении строки точек области с номером  $i$  и столбца точек области с номером  $j$ . Полагая, что каждый элемент матрицы задает цвет соответствующей точки экрана, получить на экране изображение, закодированное в матрице  $A$ .
77. Даны две целочисленные квадратные матрицы порядка  $n$ . В каждой из матриц закодировано изображение прямоугольной области экрана размером  $n \times n$  точек с координатами левого верхнего угла 0,0. Все элементы обеих матриц - это числа, равные нулю, если точка составляет фон, или единице, если точка—часть изображения. Получить на экране изображение, являющееся пересечением изображений, закодированных в первой и второй матрицах;
78. Даны две целочисленные квадратные матрицы порядка  $n$ . В каждой из матриц закодировано изображение прямоугольной области экрана размером  $n \times n$  точек с координатами левого верхнего угла 0,0. Все элементы обеих матриц - это числа, равные нулю, если точка составляет фон, или единице, если точка—часть изображения. Получить на экране изображение, являющееся объединением изображений, закодированных в первой и второй матрицах.
79. Даны натуральные числа  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ , целочисленная матрица  $[a_{ij}]_{i=1, \dots, n; j=1, \dots, n}$ . Последовательность  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$  задает координаты  $n$  точек. Матрица указывает, как соединены между собой точки:  $a_{ij}=1$ , если  $i$ -я точка соединена с  $j$ -й, и  $a_{ij}=0$  в противном случае ( $a_{ij}=a_{ji}$ ). Получить на экране точки, заданные последовательностью  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$  и соединить их между собой так, как указано в данной матрице.
80. Пусть  $A_1, A_2, A_3, \dots$  - последовательность квадратных матриц из нулей и единиц такая, что порядок матрицы  $A_i$  равен  $3^i$  и

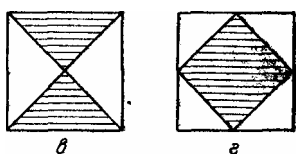
$$\bullet \quad A_i = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\bullet \quad \text{при } i > 1 \text{ имеет место } A_i = \begin{bmatrix} A_{i-1} & 0 & A_{i-1} \\ 0 & A_{i-1} & 0 \\ A_{i-1} & 0 & A_{i-1} \end{bmatrix}$$

где 0 обозначает часть матрицы, заполненную нулями. Дано натуральное число  $n$ . Построить изображение квадратной области экрана, закодированное в матрице  $A_n$ . Левый верхний угол области должен совпадать с левым верхним углом экрана. Опробовать различные способы использования цвета при построении изображения. Если фоновый цвет имеет номер 0, а остальные цвета - номера 1, ...,  $k$ , то при обработке элемента  $a_{ij}$  можно, например, брать цвет с номером  $m+1$ , где  $m$  равно остатку от деления  $i^2 + j^3$  на  $k$ .

81. Дана символьная квадратная матрица порядка  $n$ . Заменить буквой  $a$  все ее элементы, расположенные выше главной диагонали.
82. Даны натуральное  $n$ , символьная квадратная матрица порядка  $n$ . Получить последовательность  $b_1, \dots, b_n$  из нулей и единиц, в которой  $b_i = 1$  тогда и только тогда, когда в  $i$ -й строке число символов \* не меньше числа пробелов.
83. Дана символьная матрица размера  $n \times m$ . Найти номер первой по порядку строки, содержащий наибольшее число цифр.
84. Дана символьная матрица размера  $n \times m$ . Найти номер первого по порядку столбца, содержащего наименьшее число пробелов на пересечении со строками, номера которых четны.
85. Дана символьная матрица размера  $n \times m$ . Найти номер последней по порядку строки, содержащей наибольшее количество букв ш, щ.

86. Дана символьная матрица размера  $n \times m$ . Найти номер последнего по порядку столбца, в котором содержится наибольшее количество попарно различных символов.
87. При перепечатке текста на пишущей машинке часто получается так, что в конце строки остается несколько неиспользованных позиций. Число неиспользованных позиций меняется от строки к строке, и поэтому правый край отпечатанного текста получается неровным. Типографский набор дает ровный правый край, в частности, за счет увеличения промежутков между словами, встречающимися в строке. Предлагается задача выбора подходящих промежутков. Дана символьная матрица  $n \times m$ , в каждой из строк которой имеется, по крайней мере, один пробел, за которым следует отличный от пробела символ (т. е. имеется, по крайней мере, одна группа пробелов внутри строки). За счет изменения групп пробелов внутри строк надо добиться того, чтобы в конце каждой из строк пробелы отсутствовали. Количество пробелов в разных группах, располагающихся внутри одной и той же строки, должны различаться не более чем на единицу.
88. Дана действительная квадратная матрица порядка  $n$ . Найти сумму элементов первого столбца.
89. Дана действительная квадратная матрица порядка  $n$ . Найти сумму элементов главной и побочной диагоналей.
90. Дана действительная квадратная матрица порядка  $n$ . Найти наибольшее из значений элементов первой и последней строк.
91. Дана действительная квадратная матрица порядка  $n$ . Найти наименьшее из значений элементов побочной диагонали и двух соседних с ней линий.
92. Дана действительная квадратная матрица порядка  $n$ . Для данного натурального  $m$  ( $m \leq 2n$ ) найти сумму тех элементов матрицы, сумма индексов которых равна  $m$ .
93. Дана действительная квадратная матрица порядка  $n$ . Выяснить, верно ли, что наибольшее из значений элементов главной диагонали больше, чем наименьшее из значений элементов побочной диагонали.
94. Даны натуральное  $k$  и (построчно элементы квадратной вещественной матрицы  $A$   $n$ -го порядка. Вычислить  $k$ -ю степень этой матрицы ( $A^1=A$ ,  $A^2=A \cdot A$ ,  $A^3=A^2 \cdot A$  и т.д.).
95. Дана (построчно) вещественная матрица размером  $n \times m$ . Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.
96. Рассматривая элементы массива  $M$  как координаты точек на плоскости, найти  $d$  - наибольшее расстояние между этими точками.
97. Найти  $s$  - сумму элементов из заштрихованной области массива  $A$



98. Определить  $k$  - количество различных элементов массива  $S$  размерности  $n \times m$  (т.е. повторяющиеся элементы считать один раз).
99. Заполнить календарь на текущий год соответствующими днями недели (для несуществующих дат указать нет), представленный в виде массива  $12 \times 31$ .
100. Определить  $k$  - количество «особых» элементов массива  $S$  размерности  $n \times n$ , считая элемент «особым», если он больше суммы остальных элементов своего столбца.
101. Определить  $k$  - количество «особых» элементов массива  $S$  размерности  $n \times n$ , считая элемент «особым», если в его строке слева от него находятся элементы, меньшие его, а справа - большие.
102. Преобразовать массив  $S$  размерности  $n \times n$ , осуществив поворот элементов вокруг его центра на  $90^\circ$  против часовой стрелки.
103. Определить, является ли заданная целая квадратная матрица  $n$ -го порядка симметричной (относительно главной диагонали).
104. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером  $n \times m$  напечатать индексы всех ее седловых точек.

105. Дана вещественная матрица размером  $n \times n$ , все элементы которой различны. Найти скалярное произведение строки, в которой находится наибольший элемент матрицы, на столбец с наименьшим элементом.
106. Определить, является ли заданная целая квадратная матрица  $n$ -го порядка ортонормированной, т. е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1.
107. Определить, является ли заданная целая квадратная матрица  $n$ -го порядка магическим квадратом, т. е. такой, в которой суммы элементов во всех строках и столбцах одинаковы.
108. По заданным коэффициентам  $A_{ij}$  и правым частям  $b_i$ , решить систему линейных уравнений 
$$\sum_{j=1}^n A_{ij} x_j = b_i \quad (i=1, 2, \dots, n)$$
 считая, что ее определитель отличен от 0. (Рекомендация: систему предварительно привести к «треугольному» виду.)
109. Даны координаты  $n$  векторов  $n$ -мерного линейного пространства. Определить, являются ли они линейно независимыми.
110. Дана квадратная матрица  $n$ -го порядка. Найти матрицу, обратную ей, или установить, что такой не существует. (Замечание: если линейными преобразованиями строк привести заданную матрицу к единичной, то этими же преобразованиями единичная матрица будет приведена к искомой обратной матрице.)

## Файлы

1. Дан файл  $f$ , компоненты которого являются действительными числами. Найти сумму компонент файла  $f$ .
2. Дан файл  $f$ , компоненты которого являются действительными числами. Найти произведение компонент файла  $f$ .
3. Дан файл  $f$ , компоненты которого являются действительными числами. Найти сумму квадратов компонент файла  $f$ .
4. Дан файл  $f$ , компоненты которого являются действительными числами. Найти модуль суммы и квадрат произведения компонент файла  $f$ .
5. Дан файл  $f$ , компоненты которого являются действительными числами. Найти последнюю компоненту файла.
6. Дан файл  $f$ , компоненты которого являются действительными числами. Найти наибольшее из значений компонент.
7. Дан файл  $f$ , компоненты которого являются действительными числами. Найти наименьшее из значений компонент с четными номерами.
8. Дан файл  $f$ , компоненты которого являются действительными числами. Найти наибольшее из значений модулей компонент с нечетными номерами.
9. Дан файл  $f$ , компоненты которого являются действительными числами. Найти сумму наибольшего и наименьшего из значений компонент.
10. Дан файл  $f$ , компоненты которого являются действительными числами. Найти разность первой и последней компонент файла.
11. Дан файл  $f$ , компоненты которого являются целыми числами. Найти количество четных чисел среди компонент.
12. Дан файл  $f$ , компоненты которого являются целыми числами. Найти количество удвоенных нечетных чисел среди компонент.
13. Дан файл  $f$ , компоненты которого являются целыми числами. Найти количество квадратов нечетных чисел среди компонент.
14. Дано натуральное  $n$ . Записать в файл  $g$  целые числа  $b_1, \dots, b_n$ , где  $b_k = k!$ .
15. Дано натуральное  $n$ . Записать в файл  $g$  целые числа  $b_1, \dots, b_n$ , где  $b_k = 2^{k+1}$ .
16. Дано натуральное  $n$ . Записать в файл  $g$  целые числа  $b_1, \dots, b_n$ , где  $b_k = k^2$ .
17. Дано натуральное  $n$ . Записать в файл  $g$  целые числа  $b_1, \dots, b_n$ , где  $b_k = 2^k + 3^{k+1}$ .
18. Последовательность  $x_1, x_2, \dots, x_n$  образована по закону  $x_i = (i-1)/(i^3 + |\operatorname{tg}(2i)|)$ . Дано действительное  $\varepsilon > 0$ . Записать в файл  $h$  члены последовательности  $x_1, x_2, \dots$ , остановившись после первого члена, для которого выполнено  $|x_i| < \varepsilon$ .
19. Дан символьный файл  $f$ . Получить копию файла в файле  $g$ .
20. Даны символьные файлы  $f$  и  $g$ . Переписать с сохранением порядка следования компоненты файла  $f$  в файл  $g$ , а компоненты файла  $g$  - в файл  $f$ . Использовать вспомогательный файл  $h$ .
21. Даны файлы  $f_1, f_2, f_3, f_4, f_5$  компоненты которых являются действительными числами. Организовать обмен компонентами между файлами в соответствии со следующей схемой:  $f_1 \rightarrow f_3, f_2 \rightarrow f_4, f_3 \rightarrow f_5, f_4 \rightarrow f_2, f_5 \rightarrow f_1$ , т. е. компоненты файла  $f_1$  переписываются в файл  $f_3$ , компоненты файла  $f_2$  в  $f_4$  и т. д. Разрешается использовать только один вспомогательный файл  $h$ .
22. Дан символьный файл  $f$ . В файле  $f$  не менее двух компонент. Определить, являются ли два первых символа файла цифрами. Если да, то установить, является ли число, образованное этими цифрами, четным.

23. Дан файл  $f$ , компоненты которого являются целыми числами. Получить в файле  $g$  все компоненты файла  $f$  являющиеся четными числами.
24. Дан файл  $f$ , компоненты которого являются целыми числами. Получить в файле  $g$  все компоненты файла  $f$  делящиеся на 3 и не делящиеся на 7.
25. Дан файл  $f$ , компоненты которого являются целыми числами. Получить в файле  $g$  все компоненты файла  $f$  являющиеся точными квадратами.
26. Дан файл  $f$ , компоненты  $u_0, u_1, \dots, u_n$  которого являются последовательными числами Фибоначчи. Получить в файле  $f$  последовательные числа Фибоначчи  $u_0, u_1, \dots, u_{n+1}$ .
27. Дан символьный файл  $f$ . Получить файл  $g$ , образованный из файла  $f$  заменой всех его прописных (больших) букв одноименными строчными (малыми).
28. Вычислить по схеме Горнера значение многочлена с рациональными коэффициентами для данного рационального значения переменной. Считать, что числители и знаменатели коэффициентов записаны в файле  $f$ : вначале числитель и знаменатель старшего коэффициента и т. д., в последнюю очередь числитель и знаменатель свободного члена.
29. Дан файл  $f$ , компоненты которого являются целыми числами. Записать в файл  $g$  все четные числа файла  $f$ , а в файл  $h$  - все нечетные. Порядок следования чисел сохраняется.
30. Дан символьный файл  $f$ . Записать в файл  $g$  компоненты файла  $f$  в обратном порядке.
31. Даны символьные файлы  $f$  и  $g$ . Записать в файл  $h$  сначала компоненты файла  $f$ , затем - компоненты файла  $g$  с сохранением порядка.
32. Дан файл  $f$ , компоненты которого являются целыми числами. Получить файл  $g$ , образованный из файла  $f$  исключением повторных вхождений одного и того же числа.
33. Дан файл  $f$ , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл  $f$  содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл  $h$ , переписать компоненты файла  $f$  в файл  $g$  так, чтобы в файле  $g$  не было двух соседних чисел с одним знаком.
34. Дан файл  $f$ , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл  $f$  содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл  $h$ , переписать компоненты файла  $f$  в файл  $g$  так, чтобы в файле  $g$  сначала шли положительные, потом отрицательные числа.
35. Дан файл  $f$ , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл  $f$  содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл  $h$ , переписать компоненты файла  $f$  в файл  $g$  так, чтобы в файле  $g$  числа шли в следующем порядке: два положительных, два отрицательных, два положительных, два отрицательных и т. д. (предполагается, что число компонент в файле  $f$  делится на 4).
36. Дан файл  $f$ , компоненты которого являются целыми числами. Никакая из компонент файла  $f$  не равна нулю. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных, десять положительных, десять отрицательных и т. д. Переписать компоненты файла  $f$  в файл  $g$  так, чтобы в файле  $g$  числа шли в следующем порядке: пять положительных, пять отрицательных, пять положительных, пять отрицательных и т. д. (предполагается, что число компонент файла  $f$  делится на 20).
37. Дан файл  $f$ , компоненты которого являются целыми числами. Никакая из компонент файла  $f$  не равна нулю. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных, десять положительных, десять отрицательных и т. д. Переписать компоненты файла  $f$  в файл  $g$  так, чтобы в файле  $g$  числа шли в следующем порядке: двадцать положительных, двадцать отрицательных, двадцать положительных, двадцать отрицательных и т. д. (предполагается, что число компонент файла  $f$  делится на 40).
38. Дан файл  $f$ , компоненты которого являются целыми числами. Число компонент файла делится



- на 100. Записать в файл g наибольшее значение первых ста компонент файла f, затем - следующих ста компонент и т. д.
39. Дан файл f, компоненты которого являются целыми числами. Записать в файл g наибольшее значение первых ста компонент файла f, затем - следующих ста компонент и т. д. Если в последней группе окажется менее ста компонент, то последняя компонента файла g должна быть равна наибольшей из компонент файла f, образующих последнюю (неполную) группу.
40. Дан символьный файл f. Добавить в его конец символы e, n, d.
41. Дан символьный файл f. Подсчитать число вхождений в файл сочетаний ab.
42. Дан символьный файл f. Определить, входит ли в файл сочетание abcdefgh.
43. Дан символьный файл f. Подсчитать число вхождений в файл каждой из букв a, b, c, d, e и вывести результат в виде таблицы a –  $N_a$ , b –  $N_b$ , c –  $N_c$ , d –  $N_d$ , e –  $N_e$ , где  $N_a$ ,  $N_b$ ,  $N_c$ ,  $N_d$ ,  $N_e$  - числа вхождений соответствующих букв.
44. Даны символьные файлы f и g. Определить, совпадают ли компоненты файла f с компонентами файла g. Если нет, то получить номер первой компоненты, в которой файлы f и g отличаются между собой. В случае, когда один из файлов имеет n компонент ( $n \geq 0$ ) и повторяет начало другого (более длинного) файла, ответом должно быть число  $n+1$ .
45. Даны символьные файлы f и g. Записать в файл h все начальные совпадающие компоненты файлов f и g.
46. Дан символьный файл f. Записать в файл g с сохранением порядка следования те символы файла f.
47. Дан символьный файл f. Записать в файл g с сохранением порядка следования те символы файла f которым в этом файле предшествует буква a.
48. Дан символьный файл f. Записать в файл g с сохранением порядка следования те символы файла f вслед за которым в этом файле идет буква a.
49. Дан символьный файл f. Удалить из файла все однобуквенные слова (группа символов, разделенных пробелами (одним или несколькими) и не содержащих пробелов внутри себя) и лишние пробелы. Результат записать в файл g.
50. Дан символьный файл f. Найти самое длинное слово (группа символов, разделенных пробелами (одним или несколькими) и не содержащих пробелов внутри себя) среди слов, вторая буква которых есть «е». Если слов с наибольшей длиной несколько, то найти последнее. Если таких слов нет вообще, то сообщить об этом. Решить эту задачу полагая, что слова состоят не более чем из 10 символов.
51. Дан символьный файл f. Найти самое длинное слово (группу символов, разделенных пробелами (одним или несколькими) и не содержащих пробелов внутри себя) среди слов, вторая буква которых есть «е». Если слов с наибольшей длиной несколько, то найти последнее. Если таких слов нет вообще, то сообщить об этом. Решить эту задачу без ограничения на число символов в слове.
52. Дан символьный файл f. Считая, что количество символов в слове (группа символов, разделенных пробелами (одним или несколькими) и не содержащие пробелов внутри себя) не превосходит двадцати. Определить, сколько в файле f имеется слов, состоящих из одного, двух, трех и т. д. символов.
53. Дан символьный файл f. Предполагается, что длина одного слова (группа символов, разделенных пробелами (одним или несколькими) и не содержащие пробелов внутри себя) не превосходит десяти и что число слов делится на 100. Подготовить файл для печати слов в две колонки по пятьдесят строк на странице. Слова должны быть размещены в файле f<sub>1</sub> в следующем порядке: 1-е слово, 51-е слово, 2-е слово, 52-е слово, ..., 50-е слово, 100-е слово, затем (следующая страница) 101-е слово, 151-е слово, ..., 150-е слово, 200-е слово и т. д.

54. Дан символьный файл  $f$ , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия␣имя␣отчество, фамилия␣имя␣отче-ство, ... Записать эти сведения в файле  $g$ , используя образец: имя␣отчество␣фамилия, имя␣отчество␣фамилия, ...
55. Дан символьный файл  $f$ , содержащий сведения о сотрудниках учреждения, записанные по следующему образцу: фамилия␣имя␣отчество, фамилия␣имя␣отче-ство, ... Записать эти сведения в файле  $g$ , используя образец: фамилия␣и.о., фамилия␣и.о., ...
56. Дан символьный файл  $f$ , содержащий произвольный текст длиной более 5000 слов. Слова в тексте разделены пробелами и знаками препинания. Получить 100 наиболее часто встречающихся слов и число их появлений. Решить задачу без ограничения на длины слов.
57. Дан символьный файл  $f$ , содержащий произвольный текст длиной более 5000 слов. Слова в тексте разделены пробелами и знаками препинания. Получить 100 наиболее часто встречающихся слов и число их появлений. Решить задачу предполагая, что любое слово текста состоит не более чем из 16 букв.
58. Даны два символьных файла  $f$  и  $g$ . Файл  $f$  содержит произвольный текст. Слова в тексте разделены пробелами и знаками препинания. Файл  $g$  содержит не более 40 слов, которые разделены запятыми. Эти слова образуют пары: каждое первое слово считается заменяемым, каждое второе слово - заменяющим. Найти в файле  $f$  все заменяемые слова и заменить их на соответствующие заменяющие. Результат поместить в файле  $g$ .
59. Прямая на плоскости задается уравнением  $ax+by+c=0$ , где  $a$  и  $b$  одновременно не равны нулю. Будем рассматривать только прямые, для которых коэффициенты  $a, b, c$  - целые числа. Пусть  $f$  - файл, содержащий коэффициенты нескольких прямых (не менее трех). Переписать из файла  $f$  в файл  $g$  коэффициенты тех прямых, которые параллельны первой из прямых, заданных в файле  $f$ .
60. Прямая на плоскости задается уравнением  $ax+by+c=0$ , где  $a$  и  $b$  одновременно не равны нулю. Будем рассматривать только прямые, для которых коэффициенты  $a, b, c$  - целые числа. Пусть  $f$  - файл, содержащий коэффициенты нескольких прямых (не менее трех). Переписать из файла  $f$  в файл  $g$  коэффициенты тех прямых, которые перпендикулярны первой из прямых, заданных в файле  $f$ .
61. Прямая на плоскости задается уравнением  $ax+by+c=0$ , где  $a$  и  $b$  одновременно не равны нулю. Будем рассматривать только прямые, для которых коэффициенты  $a, b, c$  - целые числа. Пусть  $f$  - файл, содержащий коэффициенты нескольких прямых (не менее трех). Переписать из файла  $f$  в файл  $g$  коэффициенты тех прямых, которые пересекают первую из прямых, заданных в файле  $f$ .
62. Прямая на плоскости задается уравнением  $ax+by+c=0$ , где  $a$  и  $b$  одновременно не равны нулю. Будем рассматривать только прямые, для которых коэффициенты  $a, b, c$  - целые числа. Пусть  $f$  - файл, содержащий коэффициенты нескольких прямых (не менее трех). Переписать из файла  $f$  в файл  $g$  коэффициенты всех различных прямых.
63. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл  $f$ , содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Найти багаж, средний вес одной вещи в котором отличается не более чем на 0,3 кг от общего среднего веса вещи.
64. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл  $f$ , содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Найти число пассажиров, имеющих более двух вещей, и число пассажиров, количество вещей которых превосходит среднее число вещей.
65. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл  $f$ , содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Определить, имеются -

ли два пассажира, багажи которых совпадают по числу вещей и различаются по весу не более чем на 0,5 кг.

66. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *f*, содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Выяснить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей, и по весу.
67. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *f*, содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом не менее 30 кг.
68. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *f*, содержащий информацию о багаже нескольких пассажиров, информация о багаже каждого отдельного пассажира представляет собой соответствующую пару чисел. Дать сведения о багаже, число вещей в котором не меньше, чем в любом другом багаже, а вес вещей не больше, чем в любом другом багаже с этим же числом вещей.
69. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли в школе однофамильцы.
70. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли однофамильцы в каких-либо параллельных классах.
71. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли однофамильцы в каком-нибудь классе.
72. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли в школе однофамильцы с одним именем.
73. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли однофамильцы с одним именем в каких-либо параллельных классах.
74. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, имеются ли однофамильцы с одним именем в каком-нибудь классе.
75. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, в каких классах насчитывается более 35 учащихся.
76. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Выяснить, на сколько человек в восьмых классах больше, чем в десятых.
77. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Собрать в файле *g* сведения об учениках 9-х и 10-х классов, поместив вначале сведения об учениках класса 9а, затем 9б и т. д., затем 10а, 10б и т. д.
78. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится. Дан файл *f*, содержащий сведения об учениках школы. Получить список учеников данного класса по следующим образцам: фамилия␣имя, фамилия␣и., и.фамилия.

79. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится, отметки, полученные учениками в последней четверти. Дан файл f, содержащий сведения об учениках школы. Выяснить, сколько учеников школы не имеют отметок ниже четырех.
80. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения и буквы), в котором он учится, отметки, полученные учениками в последней четверти. Дан файл f, содержащий сведения об учениках школы. Собрать в файле g сведения о лучших учениках школы, т. е. об учениках, не имеющих отметок ниже четырех и по сумме баллов не уступающих другим ученикам своего и параллельных классов.
81. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f, содержащий сведения о нескольких автомобилях. Найти фамилии владельцев и номера автомобилей данной марки.
82. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f, содержащий сведения о нескольких автомобилях. Найти количество автомобилей каждой марки.
83. Дан файл f, содержащий различные даты. Каждая дата - это число, месяц и год. Найти год с наименьшим номером.
84. Дан файл f, содержащий различные даты. Каждая дата - это число, месяц и год. Найти все весенние даты.
85. Дан файл f, содержащий различные даты. Каждая дата - это число, месяц и год. Найти самую позднюю дату.
86. Дан файл f, содержащий сведения о книгах. Сведения о каждой из книг - это фамилия автора, название и год издания. Найти названия книг данного автора, изданных с 1960 г.
87. Дан файл f, содержащий сведения о книгах. Сведения о каждой из книг - это фамилия автора, название и год издания. Определить, имеется ли книга с названием «Информатика». Если да, то сообщить фамилию автора и год издания. Если таких книг несколько, то сообщить имеющиеся сведения обо всех этих книгах.
88. Дан файл f, который содержит номера телефонов сотрудников учреждения: указывается фамилия сотрудника, его инициалы и номер телефона. Найти телефон сотрудника по его фамилии и инициалам.
89. Дан файл f, содержащий сведения о кубиках: размер каждого кубика (длина ребра в сантиметрах), его цвет (красный, желтый, зеленый или синий) и материал (деревянный, металлический, картонный). Найти количество кубиков каждого из перечисленных цветов и их суммарный объем.
90. Дан файл f, содержащий сведения о кубиках: размер каждого кубика (длина ребра в сантиметрах), его цвет (красный, желтый, зеленый или синий) и материал (деревянный, металлический, картонный). Найти количество деревянных кубиков с ребром 3 см и количество металлических кубиков с ребром, большим 5 см.
91. Дан файл f, содержащий сведения о веществах: указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор). Найти удельные веса и названия всех полупроводников.
92. Дан файл f, содержащий сведения о веществах: указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор). Выбрать данные о проводниках и упорядочить их по убыванию удельных весов.
93. Дан файл f, содержащий сведения об экспортируемых товарах: указывается наименование товара, страна, импортирующая товар, и объем поставляемой партии в штуках. Найти страны, в которые экспортируется данный товар, и общий объем его экспорта.
94. Даны два файла f и g. Файл f - это инвентарный файл, содержащий сведения о том, сколько изделий каких видов продукции хранится на складе (вид продукции задается его порядковым

номером). Файл g - это вспомогательный файл, содержащий сведения о том, на сколько уменьшилось или увеличилось количество изделий по некоторым видам продукции. Вспомогательный файл может содержать несколько сообщений по продукции одного вида или не содержать ни одного такого сообщения. Обновить инвентарный файл на основе вспомогательного, образовав новый файл h.

95. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: названия игрушек, цена которых не превышает m руб. и которые подходят детям n лет.
96. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: цену самого дорогого конструктора, оформленную по образцу ... руб. ... коп.
97. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: названия наиболее дорогих игрушек (цена которых отличается от цены самой дорогой игрушки не более чем на k руб.).
98. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: названия игрушек, которые подходят как детям m лет, так и детям n лет.
99. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: цены всех кубиков, оформленные по образцу ... руб. ... коп.
100. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: можно ли подобрать игрушку, любую, кроме мяча, подходящую ребенку n лет, и дополнительно мяч так, чтобы суммарная стоимость игрушек не превосходила m руб.?
101. Дан файл f, содержащий сведения об игрушках: указывается название игрушки (например, кукла, кубики, мяч, конструктор и т. д.), ее стоимость в копейках и возрастные границы детей, для которых игрушка предназначена (например, для детей от двух до пяти лет). Получить следующие сведения: имеется ли мяч ценой m руб. k коп., предназначенный детям от n1 до n2 лет?; если нет, занести сведения об этой игрушке в файл f.
102. Даны натуральное k, символьный файл f и текстовый файл g. Файл f содержит n слов, каждое из которых будем называть ключевым. Сформировать файл h, который содержит строки файла g, циклически сдвинутые так, чтобы каждое ключевое слово, входящее в строку, началось с k-ой позиции. Строки, не содержащие ключевых слов, в файл h не включаются. Строки, которые содержат m ключевых слов, записываются в файл h m раз.
103. Дан текстовый файл f, содержащий программу на языке паскаль. Проверить эту программу на несоответствие числа открывающих и закрывающих круглых скобок. Считать, что каждый оператор программы занимает не более одной строки файла f.
104. Дан текстовый файл f, содержащий программу на языке паскаль. Проверить эту программу на несоответствие числа открывающих и закрывающих круглых скобок. Считать, что каждый оператор программы может занимать произвольное число строк файла f.

105. Дан текстовый файл *f*. Получить все его строки, содержащие более *k* символов.
106. Дан текстовый файл *f*. Переписать в файл *g* все компоненты файла *f* с заменой в них символа 0 на символ 1 и наоборот.
107. Дан текстовый файл *f*. Получить самую длинную строку файла. Если в файле имеется несколько строк с наибольшей длиной, то получить одну из них.
108. Дан текстовый файл *f*. Записать в перевернутом виде строки файла *f* в файл *g*. Порядок строк в файле *g* должен совпадать с порядком исходных строк в файле *f*.
109. Дан текстовый файл *f*. Записать в перевернутом виде строки файла *f* в файл *g*. Порядок строк в файле *g* должен быть обратным по отношению к порядку строк исходного файла.
110. Дан текстовый файл *f*. Переписать компоненты файла *f* в файл *g*, вставляя в начало каждой строки по одному пробелу. Порядок компонент должен быть сохранен.
111. Даны текстовый файл, строка *s*. Получить все строки файла *f*, содержащие в качестве фрагмента строку *s*.
112. Дан текстовый файл *f*. Исключить пробелы, стоящие в концах его строк. Результат поместить в файл *g*.
113. Даны два текстовых файла *f* и *g*. Определить, совпадают ли компоненты файла *f* с компонентами файла *g*. Если нет, то получить номер первой строки и позицию первого символа в этой строке, в которых файлы *f* и *g* отличаются между собой. В случае, когда один из файлов имеет *n* компонент ( $n \geq 0$ ) и повторяет начало другого (более длинного) файла, ответом должно быть число  $n+1$ .
114. Дан файл *f*, компоненты которого являются натуральными числами. Количество чисел в файле кратно 4. Первые два числа из каждых четырех задают координаты левого верхнего угла прямоугольника, следующие два числа - координаты его правого нижнего угла. Построить прямоугольники, заданные в файле *f*.
115. Дан текстовый файл *f*. Каждая строка файла содержит несколько натуральных чисел в их символьном представлении. Числа разделяются запятыми или пробелами и определяют вид некоторой геометрической фигуры, ее размеры и положение на экране. Приняты следующие соглашения: 1) отрезок прямой задается координатами своих концов, имеет номер 1; 2) прямоугольник задается координатами левого верхнего и правого нижнего угла, имеет номер 2; 3) окружность задается координатами центра и радиусом, имеет номер 3; 4) ломаная задается количеством ее вершин, их координатами и имеет номер 4. Так, например, строка 1, 10, 10, 30, 30 определяет отрезок прямой с координатами концов (10, 10) и (30, 30), а строка 3, 100, 100, 50 - окружность с центром в точке (100, 100) и радиусом 50. Построить на экране все геометрические фигуры, заданные в файле *f*.
116. Дан файл *f*, компоненты которого являются натуральными числами. Число компонент файла кратно четырем. Каждые две последовательные компоненты определяют координаты двух точек. Считая, что заданы координаты концов отрезков, построить все такие отрезки.
117. Дан файл *f*, компоненты которого являются натуральными числами. Число компонент файла кратно четырем. Каждые две последовательные компоненты определяют координаты двух точек. Считая, что заданы координаты противоположных углов прямоугольника, построить все такие прямоугольники.
118. Дан файл *f*, компоненты которого являются натуральными числами. Число компонент файла кратно четырем. Каждые две последовательные компоненты определяют координаты двух точек. Считая, что заданы координаты центра окружности и одной из ее точек, построить все такие окружности.
119. Дан файл *f*, с описанием структуры после которой идут записи. Написать программу поиска нужной записи.
120. Дан файл *f*, с описанием структуры после которой идут записи. Написать программу до-

бавления и удаления записей в файле.

121. Напишите программу, которая вычисляет среднее значение и среднеквадратичное отклонение целых чисел, записанных в файле. Результат должен иметь тип `float`. Вывод должен быть оформлен с соответствующими заголовками, значения должны быть выведены с точностью до двух знаков после запятой. Формула для вычисления среднего значения для набора целых чисел состоит в сложении всех этих чисел и делении полученной суммы на их количество. Для вычисления среднеквадратичного отклонения для набора целых чисел нужно из каждого числа вычесть среднее значение (может получиться отрицательный результат) и возвести каждую разность в квадрат. Затем сложить все эти квадраты, разделить на общее количество чисел минус 1 и затем извлечь квадратный корень из результата.

## **Записи.**

### **1. Календарь.**

Создать базу данных Date. Каждая запись – это число, месяц, год. Найти: год с наименьшим номером, все весенние даты, самую позднюю дату, все праздничные даты.

### **2. Переиндексация.**

Написать процедуру, добавления в индексированный массив нового индекса, при добавлении в файл базы данных новой записи.

### **3. Знаки зодиака.**

Запись состоит из полей: имя, день, месяц, год. Написать программу, выводящую на экран имена людей, родившихся под указанным знаком зодиака.

### **4. Ученики.**

Запишите объявление записи, содержащей следующую информацию об ученике:

Имя (строка символов)

Номер социального страхования (строка символов)

Курс (freshman, sophomore, junior, senior)

Средняя оценка (вещественное число)

Пол(M,F)

Объявите переменную предложенного типа и напишите фрагмент программы для вывода информации, содержащейся в каждом члене переменной.

Объявите массив roll из 3000 записей типа, описанного в пункте.

### **5. Запчасти.**

Напишите фрагмент программы, который считывает набор номеров запчастей и ассоциированных с ними цен. Сортируйте данные по номеру запчасти по мере их считывания. Используйте массив записей из двух членов, number и cost, для представления каждой пары входных значений. Предполагается, что ввод завершается при наступлении конца файла.

### **6. Студенты.**

Запишите объявление иерархической записи, содержащей следующую информацию о студенте:

Имя (до 30 символов)

Идентификационный номер студента

Количество прослушанных часов до настоящего момента

Оценки за курсы (список, состоящий не более, чем из 50 элементов; каждый элемент содержит целочисленный номер курса и последнюю оценку за него)

Дата зачисления (месяц и год)

Курс (freshman, sophomore, junior, senior)

Средняя оценка

Каждая запись и перечисляемый тип должны иметь отдельное объявление.

Напишите программу сортирующую записи по курсам в алфавитном порядке, идентификационным номерам.

### **7. Архитектура.**

Объявите тип «запись» с именем AptType для нужд архитектурской конторы. Тип должен отражать следующую информацию:

Землевладелец (строка не более, чем из 20 символов)

Адрес (строка не более, чем из 20 символов)

Число комнат (целое)

Цена (число с плавающей точкой)

Объявите массив available из 200 записей типа AptType.

Напишите функцию, которая считывает значения в члены переменной типа AptType. (Переменная типа «запись» должна передаваться в качестве параметра.) Порядок считывания данных тот же, что и у компонентов записи.

### **8. Библиотека.**



Вы разрабатываете систему автоматизированного учета для библиотеки. Библиотека насчитывает 50000 книг. Для каждой книги должна присутствовать запись, содержащая номер (до 10 символов), число экземпляров книги (целое число), автора книги (до 30 символов), название книги (до 100 символов) и описание содержания книги (до 300 символов). Запишите объявления типов, необходимых для представления этой информации. Подсчитайте, сколько памяти (в символах) необходимо, чтобы содержать всю информацию по библиотеке. (Предположите, что одно целое число занимает место в памяти, эквивалентное четырем символам.). Сколько записей о книгах можно поместить в памяти размером 650000.символов? Напишите функцию, которая считывает информацию о книге в запись объявленного типа. Затем напишите еще одну функцию, которая выводит содержимое записи. Запись должна передаваться в качестве параметра каждой из этих функций.

## **9. Подписка.**

Вы пишете систему автоматизированного учета подписки на журнал. Для каждого подписчика система должна хранить следующую информацию:

Имя (имя, фамилия)

Адрес (улица, город, область, почтовый индекс)

Прекращение действия подписки (месяц, год)

Дата, когда было отправлено уведомление об окончании подписки (месяц, день, год)

Количество отправленных уведомлений

Количество лет, в течение которых человек является подписчиком журнала

(0 - если новая подписка еще не была оформлена; в противном случае -1, 2 или 3 года)

Можно ли включить имя подписчика в список для других компаний

Напишите объявление иерархической записи для представления описанной информации. Каждая запись более низкого уровня должна быть объявлена отдельно как именованный тип данных.

## **10. Терминалы.**

Вы пишете программу для учета терминалов, подключенных к компьютеру компании. Компьютер может поддерживать до 30 терминалов. Для каждого из терминалов должна храниться следующая информация:

Название и модель (строка не более, чем из 15 символов)

Скорость передачи данных (целое число от 10 до 19200 символов в секунду)

Четность (перечисляемый тип, представляющий: Четно, Нечетно, Один, Ноль или Нет)

Дуплексность (перечисляемый тип, представляющий: Полу-дуплекс, Полный дуплекс)

Биты данных (целое число со значением 7 или 8)

Стоп-биты (целое число со значением 1 или 2)

Разработайте структуру данных для данной задачи и напишите объявления для всех типов данных, необходимых для реализации разработанной Вами структуры.

## **11. Строки.**

Напишите объявление записи для хранения строки не более, чем из 20 символов, и длины строки. Затем напишите функцию, возвращающую длину строки, содержащейся в этой записи. Напишите функцию, которая конкатенирует (соединяет) две строки в третью строку.

## **12. Инвентаризация.**

Вы пишете программу для управления складом компании. Для каждого товара должна храниться следующая информация:

Номер товара

Название товара

Цена

Количество

Напишите функцию и сделайте необходимые объявления для считывания переменной типа «запись» с именем Inventory.

## **13. Учет зарплаты.**

Компания Emerging Manufacturing только что установила свой первый компьютер и наняла вас в качестве программиста. Вашей первой задачей является считать данные по зарплате рабочих и вывести два отчета: (1) отчет по ошибкам и (2) отчет по зарплате. Последний должен содержать строку для каждого рабочего, а затем строку с общей суммой в конце отчета.

Ввод:

Файл данных по филиалам Набор из пяти пар номер/название филиала. Для каждого рабочего отводится одна строка. В ней хранится: идентификационный номер, номер филиала и количество отработанных часов

Эти данные отсортированы по идентификационному номеру.

Основной файл данных

Идентификационный номер

Имя

Ставка зарплаты в час

Число подчиненных

Тип работы (1 - это управление, 0 - это профсоюз)

Филиал, в котором проходит работа

Пол(M,F)

Этот файл упорядочен по идентификационным номерам.

Обратите внимание:

(1) Члены профсоюза, в отличие от управляющего персонала, за сверхурочные часы (после 40 часов) получают платы в полтора раза больше.

(2) Формула для вычисления налогов имеет вид: Если число подчиненных - 1, то ставка налога равняется 15%. В противном случае, ставка налога вычисляется по формуле

$$\left[ 1 - \left( \frac{\text{число подчиненных}}{\text{число подчиненных} + 6} \right) \right] * 15\%$$

Вывод:

Отчет по ошибкам

Содержит введенные строки, для которых нет соответствующей записи в основном файле данных, либо номера филиалов, на которых заняты рабочие, не совпадают.

Содержит общее число записей, обработанных правильно в процессе выполнения.

Отчет по зарплате

Содержит строку для каждого рабочего, содержащую имя, идентификационный номер рабочего, филиал, общая зарплата и зарплата за вычетом налогов. Содержит строку, иллюстрирующую общую сумму зарплат и сумму зарплат за вычетом налогов.

#### 14. Учет на складе

Компания Emerging Manufacturing решила использовать свой новый компьютер для учета состояния склада. Вам поручили написать программу, которую следует запускать каждую ночь. Она обрабатывает результаты работы склада за день и выводит обновленный отчет, который раздается утром менеджерам во всех пяти филиалах. Обратите внимание, что вам не сказано обновлять файл.

Ввод:

файл данных т филиалам. Набор из пяти пар номер/название филиала. Каждая строка содержит идентификационный номер товара, номер филиала и количество проданных или купленных товаров (отрицательное число обозначает, что товар был продан).

Эти данные отсортированы по идентификационному номеру товара.

Основной файл данных

Идентификационный номер товара

Название товара (без пробелов)

Количество товара в наличии

Точка упорядочения (минимальное количество наличного товара, которое приведет в переупорядочению списка).

Филиал

Этот файл также упорядочен по номеру филиала в рамках номера товара. Если товара нет в основном файле, а запись о продаже присутствует, то должно быть выведено сообщение об ошибке. Если произведена покупка, то товар должен быть добавлен в список на соответствующее место. Обратите внимание, что для каждого товара в каждом из филиалов присутствует отдельная запись.

Вывод:

Отчет по ошибкам

Содержит сообщения об ошибках

Содержит список товаров, требуемых переупорядочения (товары, для которых наличное количество меньше точки упорядочения)

Отчет по всем товарам из основного файла данных

Содержит номер товара

Содержит название товара

Содержит название филиала

Содержит количество товара в наличии

Помните, что этот отчет предназначен для менеджеров. Убедитесь, что он составлен в удобной для чтения форме.

## 15. Агентство знакомств.

Вам поручили написать программу для электронного агентства знакомств. Клиенты сообщают вам свое имя, номер телефона и список интересов. В Вашу задачу входит пополнение списков мужчин и женщин, а также поиск подходящих пар.

Структуры данных:

Постановка задачи требует создания двух списков - один для мужчин, другой для женщин. Списки должны содержать следующую информацию: имя (20 символов), номер телефона (8 символов), количество интересов (максимально 10), сами интересы (на каждый из них отводится 10 символов; должны стоять в алфавитном порядке), а также переменную, содержащую информацию о совпадении клиента (-1, если совпадений нет). При добавлении в список нового клиента, его или ее имя записывается в конец соответствующего списка. (Вам не нужно хранить имена в алфавитном порядке.)

Ввод:

Количество клиентов в настоящий момент Пол ('M' или 'F'), имя (20 символов), номер телефона (8 символов), количество интересов, список интересов (по 10 символов на каждое название, без пробелов; интересы разделяются пробелами). Такая запись присутствует для каждого клиента. Остаток файла содержит строки данных, следующего вида (все строки начинаются с 10-символьного слова, как показано ниже; символ  $\cup$  обозначает пробел):

NewClient $\cup$ пол ('M' или 'F'), имя (20 символов), количество интересов, интересы (по 10 символов на каждый; см. описание выше) Если появляется ключевое слово NewClient, вы должны добавить клиента в соответствующий список, сохранив информацию в определенных членах записи. Сравните его или ее с клиентами противоположного пола. (Пара считается совпавшей, если совпадают по крайней мере три интереса. Интересы отсортированы, что упрощает процесс поиска совпадений. Используйте сортировку вставками для упорядочения интересов.). После этого не забудьте отметить, получилось ли для клиента совпадение. Выведите имя нового клиента, совпадение для него или для нее и оба номера телефона. Если не произошло совпадения, выведите соответствующее сообщение.

(OldClient $\cup$ имя (20 символов)

Отменить совпадение для данного клиента, присвоив соответствующей переменной значение -1.

PrintMatch Вывести список всех совпадающих пар.

PrintNot $\cup\cup$

Вывести имена и номера телефонов клиентов, для которых в настоящее время нет пары.

StopProg $\cup\cup$

Это должно быть последней строкой в файле.

Вывод:

Информация, описанная выше, должна быть выведена с соответствующими обозначениями.

## 16. Разреженная матрица.

Разреженная матрица - это такая матрица (двумерный массив), в которой преобладающее большинство элементов равны нулю. Ее неэффективно хранить в виде двумерного массива, поскольку большинство элементов не содержат никакой полезной информации. Вместо этого, неравные нулю элементы должны храниться в виде массива записей, в которых первые два члена содержат номер строки и столбца, а третий член - значение элемента. Например, такая матрица:

```
0.0 0.0 7.0 0.0 0.0
0.0 0.0 0.0 0.0 8.0
0.0 0.0 0.0 0.0 0.0
```

будет храниться следующим образом:

```
1 3 7.0
2 5 8.0
```

Напишите программу, которая считывает разреженную матрицу и преобразует ее в массив записей описанного вида. В конце программа должна выводить значения, содержащиеся в записях (обозначенные соответствующим образом).

## 17. Бейсбольная команда.

Местная бейсбольная команда решила подсчитать свои рекорды. Вам поручили написать программу, которая подсчитывает средние значения очков. В команде 20 игроков, у каждого из них есть идентификационный номер от 1 до 20. Их личные очки записаны в файле в следующем виде. Каждая строка содержит четыре числа: идентификационный номер игрока и количество удачных ударов, пробежек и промахов, которые он сделал в определенной игре. Вот пример записи из файла: 3 2 1 1

Эта запись показывает, что в течение игры, игрок номер 3 бил четыре раза, из которых 2 были удачными, 1 пробежка и 1 промах. Для каждого игрока в файле есть несколько записей. Среднее количество очков вычисляется суммированием числа всех удачных ударов игрока и делением этого значения на общее число ударов. При вычислениях, пробежка не считается удачным ударом, а также не учитывается при подсчете общего числа ударов. Ваша программа должна выводить таблицу, содержащую номер каждого игрока, среднее значение очков и количество пробежек.

## 18. Рекламная компания.

Рекламная компания хочет послать всем своим клиентам письма, сообщающие о новом графике оплаты. Имена клиентов внесены в несколько различных списков компании. Все списки были объединены в один файл с именем `clientNames`, но компания не хочет посылать клиентам письма по два или три раза. Напишите программу, которая удаляет повторяющиеся записи о клиентах. В каждой строке файла записан четырехзначный код, за которым следует пробел и имя клиента. Например: 0231 Amalgamated Steel. Ваша программа должна выводить код и имя каждого клиента, но только по одному разу. Не должно происходить дублирования информации.

## 19. Реклама банка.

Один из местных банков собирается организовать большую рекламную кампанию, и ему требуется знать, сколько времени клиенты ждут обслуживания. Несколько служащих попросили вести точный учет клиентов в течение 24 часов. Собранная информация (записанная в файле) состоит из времени прибытия клиента (в часах, минутах и секундах), времени, когда клиент был обслужен, и идентификационного номера кассира. Напишите программу, которая выполняет следующие задачи:

- Считывает данные по времени ожидания.
- Вычисляет время ожидания в секундах.
- Вычисляет среднее значение и среднеквадратичное отклонение и диапазон.
- Печатает резюме, иллюстрирующее значения, рассчитанные в пункте (в).

*Ввод:*

Первая строка содержит заголовок.

Каждая из остальных строк содержит номер кассира, время прибытия и время обслуживания.

Время разбито на целое число часов, минут и секунд, в соответствии с 24-м режимом.

*Обработка:*

Вычисление среднего значения и среднеквадратичного отклонения. Определить самое короткое и самое длинное время ожидания из 100 случаев.

*Вывод:*

Вводимые данные (отображение на экране). Заголовок.

Следующие значения, имеющие соответствующие надписи: количество записей, среднее значение и среднеквадратичное отклонение, диапазон (минимальное и максимальное значения).

## 20. Класс студентов.

У вашего профессора по истории столько студентов в классе, что для нее становится проблемой следить за общей успеваемостью во время сессии. Она узнала, что вы - специалист в области программирования и попросила вас написать программу для простого статистического анализа экзаменационных оценок. Программа должна работать для класса любого размера от 0 до 100 студентов ( $0 < N < 100$ ). Напишите и протестируйте программу, которая выполняет следующие задачи:

а). Читает оценки из файла inData.

б). Вычисляет среднее значение и среднеквадратичное отклонение и процентное содержание оценок, лежащих в следующих интервалах: <10, 10-19, 20-29, 30-39,..., 80-89 и >90.

в). Печатает резюме, содержащее среднее значение оценок и среднеквадратичное отклонение, а также гистограмму, иллюстрирующую процентное распределение оценок за экзамены.

*Ввод:*

Первая строка данных содержит количество экзаменов, по которому проводится анализ, а также заголовок для отчета. Остальные строки содержат по десять оценок в каждой, и последняя строка содержит от одной до десяти оценок. Все оценки представлены целыми числами.

*Вывод:*

Входные данные в исходном виде.

Отчет, состоящий из заголовка (считанного из входного файла), количества оценок, среднего значения, среднеквадратичного отклонения (подписанных соответственно) и гистограммы.

## 21. Почтовая доставка.

Небольшая почтовая система доставляет посылки в пределах вашего города. Доставка имеет следующие ограничения:

а). Вес посылки не должен превышать 50 фунтов.

б). Размеры посылки не должны превышать 3 фута по длине, ширине или высоте, а ее «обхват» должен быть меньше 6 футов. «Обхват» посылки - это окружность, описанная вокруг двух самых коротких сторон посылки;

математическая формула для его вычисления имеет следующий вид:

«Обхват» =  $2 * (s1 + s2 + s3 - largest)$

Здесь *largest* - это самый большой из трех линейных размеров, *s1*, *s2* и *s3*. Ваша программа должна обрабатывать файл, содержащий информацию о транспортировке каждой посылки в течение недели. Каждая запись в файле содержит номер транзакции, за которым следует вес и размеры посылки (размеры могут быть указаны в любом порядке). Программа должна выводить номер транзакции, вес и почтовый сбор для каждой из принятых и не принятых посылок. В конце отчета программа должна выводить количество доставленных и не принятых посылок,

*Ввод:*

Таблица информации о посылках - вес и стоимость (содержит 25 пар значений). Эта таблица должна сохраняться в одномерных массивах. Вы можете определить стоимость доставки каждой посылки, проведя поиск сначала по массиву *weight* (веса), а затем считать значение из соответствующего элемента массива *cost* (стоимости). Если вес посылки оказывается между двумя категориями в таблице, то программа должна использовать стоимость для большего из весов.

Файл транзакций - номер транзакции, вес и три размера посылки. Число транзакций может быть любым. Предположите, что все веса являются целыми числами, и что все размеры задаются с точностью до ближайшего дюйма.

*Вывод:*

Первая строка - соответствующие заголовки.

Следующие n строк - номер транзакции, принята посылка или нет, вес и стоимость.

Последняя строка - количество доставленных и не принятых посылок.

## **22. Экзамен по психологии.**

Ваш последний экзамен по психологии - это тест из 30 вопросов. На каждый вопрос предлагается несколько вариантов ответов. Учитель пообещал, что если вы напишете программу обработки результатов тестирования, то вам не придется сдавать этот экзамен.

*Ввод:*

Первая строка содержит ключ к тесту. Правильные ответы записаны в первых 30 символах. После этого записано целое число, которое означает количество студентов, сдававших экзамен (n).

Следующие n строк содержат ответы студентов в первых 30 символах каждой строки, и имя студента в следующих 10 символах.

*Вывод:*

Для каждого студента выводится имя, количество правильных ответов и слово PASS, если студент набрал 60 или более процентов правильных ответов;

в противном случае для студента выводится слово FAIL.

## **23. Прыжки в воду.**

В соревнованиях по прыжкам в воду, каждый участник делает три прыжка различной степени сложности. Девять судей ставят за каждый прыжок оценки от 0 до 10 с интервалом в 0.5. Общая оценка за прыжок получается следующим образом: отбрасываются самая высокая и самая низкая оценки; затем остальные оценки складываются и результат умножается на коэффициент сложности. Участники выступают по очереди, и после окончания соревнования им присуждаются места в зависимости от их оценок. Напишите программу, которая выполняет описанные выше действия.

*Ввод:*

Количество участников соревнования

Имя участника (10 символов), сложность (типа float) и оценки судей (девять чисел типа float)

Информация для каждого прыжка, каждого из участников хранится в отдельной строке. Сначала следуют все данные по первому прыжку, затем по второму и третьему.

*Вывод:*

Входные данные должны быть отображены на экране, в форме таблицы с соответствующими заголовками. Например: Имя, Сложность, оценки судей (1-9) Таблица, содержащая следующую информацию, отсортированная по окончательным оценкам в убывающем порядке (победитель выводится в первой строке):

Имя Прыжок 1 Прыжок 2 Прыжок 3 Общий счет

В данной таблице Имя - это имя участника соревнования; Прыжок 1, Прыжок 2 и Прыжок 3 - это общие оценки за каждый из прыжков, вычисленные по приведенной выше схеме; Общий счет - это суммарная оценка по всем прыжкам.

## **24. Бридж.**

Колоду игральных карт можно представить в виде массива записей, где первое поле отвечает за достоинство карты, а второе поле - за ее масть. Считайте сдачу для бриджа (13 карт) и определите, стоит игроку пасовать или торговаться. Каждая карта вводится в отдельной строке, сначала масть, а затем достоинство.

Решение о том, пасовать или торговаться, основывается на количестве очков, находящихся на руках у игрока. Очки распределяются следующим образом:

Туз - 4 очка

Король - 3 очка

Дама - 2 очка

Валет - 1 очко

Требуется получить сумму очков, находящихся на руках после сдачи, и вывести одно из следующих сообщений:

Если сумма очков меньше 13 - «Пасовать»

Если сумма лежит в пределах от 13 до 16 - «Торговаться до одной масти»

Если сумма лежит в пределах от 17 до 19 - «Торговаться до одной без козыря»

Если сумма лежит в пределах от 20 до 22 - «Торговаться до одной масти»

Если сумма очков больше 22 - «Торговаться до пары карт одной масти»

## 25. Информация по товарам

Информация по товарам содержит данные о магазине, отделе и месяце продажи. Напишите функции для вычисления и вывода общего количества товаров, проданных за год в каждом отделе каждого магазина, конкретного товара по месяцам, конкретного товара по отделу за год. Данные хранятся в массиве записей. Массив с данными должен передаваться в качестве параметра функциям.

## 26. Собрание акционеров.

Только уполномоченные держатели акций могут посещать собрания акционеров. Напишите программу, которая считывает с клавиатуры имя человека, сверяет его со списком держателей акций и выводит сообщение о том, имеет ли право данный человек посещать собрание акционеров. Список держателей акций находится в файле в следующем формате: № акции (6 цифр), пробел, имя (максимум 10 символов), пробел, фамилия (максимум 20 символов). Используйте признак конца файла для прекращения чтения данных из файла. Максимальное количество держателей акций равняется 1000.

Для пользователя должно быть выведено приглашение ввести его (ее) имя в таком же формате, как оно хранится в файле. Если такого имени в списке нет, программа должна снова вывести приглашение к вводу. Сообщение о том, что человек с таким именем не может посещать собрание, должно выводиться только после двойной проверки. Приглашение должно содержать сведения о том, что для окончания работы достаточно нажать букву Q. Распечатайте отчет, иллюстрирующий количество акционеров на момент проведения собрания, сколько из них присутствовали на собрании, скольким было отказано в праве участвовать в собрании. В соответствии с этим отчетом, определите список имен акционеров, в котором после каждого имени запишется либо слово «*Presents*», либо слово, «*Absent*».

## 27. Взносы клиентов.

Страховая компания наняла вас для написания программы печати списка их клиентов и взносов, которые делает каждый клиент. Размер взноса зависит от возраста, который имел клиент на тот момент, когда начал пользоваться услугами компании. Следующая таблица используется для вычисления взноса для каждого клиента, но эти ставки довольно часто меняются.

Возраст	Размер взноса
25	\$277
35	\$287
45	\$307
55	\$327
65	\$357

Каждое значение возраста, указанное в таблице - это верхняя граница интервала, которому соответствует указанный размер взноса. Например, если клиентка заключила контракт, когда ей было 37 лет, то она должна заплатить \$307.00. Напишите программу, которая считывает таблицу в два параллельных массива, а затем считывает имена клиентов и их возраст на момент заключения контракта в виде массива записей. Таблица и информация об именах и возрасте клиентов хранятся в двух файлах. Напечатайте форматированный список с соответствующими заголовками, иллюстрирующий имя каждого клиента, его (ее) возраст на момент заключения контракта, а также требуемый размер страхового взноса.

## 28. Торговые агенты.

Компания хочет знать процентные соотношения суммарных продаж и расходов, применительно к каждому торговому агенту. Каждому из них соответствует запись, состоящая из несколь-

ких строк. В первой строке содержится его (ее) фамилия (максимум 20 символов), запятая и затем его (ее) имя (максимум 10 символов). В следующей строке содержится его (ее) суммарные продажи (тип `int`) и затраты (тип `float`). Напишите программу, которая генерирует отчет следующего вида. В строке заголовка содержится общие продажи и затраты. За заголовком следует таблица с именами, фамилиями, процентом от общих продаж и процентом от общих затрат для каждого торгового агента.

## **29. Школа и спорт.**

В файле хранятся записи, состоящие из № школы, названия вида спорта которым занимаются студенты, количества денег выделенных на развитие данного спорта и количества учеников им занимающихся. Написать программу позволяющую:

- а). Определить, какая из школ потратила максимальное количество денег на футбол.
- б). Определить, на какой вид спорта последняя школа потратила больше всего денег.
- в). Определить, в какой школе больше народу играет в баскетбол.
- г). Определить, каким видом спорта занимается большинство учеников в третьей школе.
- д). Определить сумму, потраченную всеми школами на волейбол.
- е). Найти общее количество студентов, занимающихся спортом. (Предполагается, что каждый студент занимается одним видом спорта.)
- ж). Определить, в какой из школ больше студентов занимается спортом.
- з). Определить самый популярный вид спорта с точки зрения затраченных денег.
- и). Определить самый популярный вид спорта с точки зрения количества занимающихся им студентов.

## **30. Выборы.**

Предварительный опрос на улице был оформлен в виде анкеты, в которой было предложено ответить на 5 вопросов. Необходимо было заполнить следующие графы: возраст, пол, род занятий, регион, кандидат за которого будет голосовать. Написать программу обработки общественного мнения позволяющую:

- а) Определить общий рейтинг кандидата.
- б) Определить рейтинг кандидата среди мужчин.
- в) Определить рейтинг кандидата среди женщин.
- г) Определить рейтинг кандидата по регионам.
- д) Определить рейтинг кандидата среди возрастных категорий.
- е) Определить рейтинг кандидата среди социальных групп.

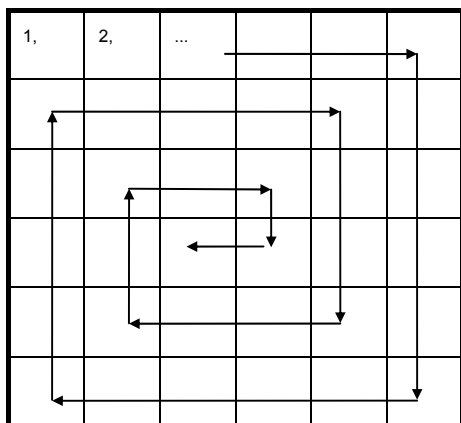


## Алгоритмы.

1. В данной последовательности  $N$  вещественных чисел выбрать возрастающую подпоследовательность наибольшей длины.
2. Напечатать все простые числа, не превосходящие заданное число  $M$ .
3. Напечатать все четырехзначные натуральные числа, в десятичной записи которых нет двух одинаковых цифр. Обобщить эту задачу на  $n$ -значные числа.
4. Сократить дробь. Даны натуральные числа  $M, N$ . Найти такие натуральные числа  $M_1, N_1$ , не имеющие общих делителей, что  $M_1/N_1 = M/N$ .
5. Период дроби. Ввести натуральные числа  $M$  и  $N$  и напечатать десятичный период дроби  $M/N$ . Например, для дроби  $1/7$  период будет (142857), а если дробь конечная, то ее период состоит из одной цифры 0.
6. В написанном выражении  $(((((1?2)?3)?4)?5)?6)$  вместо каждого знака  $?$  вставить знак одной из четырех арифметических операций  $+, -, *, /$  так, чтобы результат вычислений равнялся 35 (при делении дробная часть в частном отбрасывается). Достаточно найти одно решение.
7. Можно ли заданное натуральное число  $M$  представить в виде суммы двух квадратов натуральных чисел? Написать программу решения этой задачи.
8. Составить программу вывода всех трехзначных десятичных чисел, сумма цифр которых равна данному целому числу.
9. Найти минимальное число, которое представляется суммой четырех квадратов натуральных чисел не единственным образом.
10. Дана матрица, состоящая из нулей и единиц. Найти наибольший по площади прямоугольник, состоящий из одних единиц.
11. Даны два массива  $A$  и  $B$  из целых чисел. Напечатать в порядке возрастания все суммы вида  $A[i] + B[j]$ .
12. Задан массив  $A(M)$  из  $M$  попарно различных целых чисел. Напечатать все перестановки этих чисел.
13. Дан двумерный целочисленный массив  $A(2, 15)$ . Известно, что среди его элементов два и только два равны между собой. Напечатать их индексы.
14. Дано натуральное число  $M$  и целочисленный массив  $A(M)$ . Сосчитать и напечатать, сколько различных чисел в этом массиве. Например, в массиве 5, 7, 5 различных чисел два (5, 7).
15. В массиве  $X(M, N)$  все числа различны. В каждой строке выбирается минимальный элемент, затем среди этих чисел выбирается максимальное. Напечатать номер строки массива  $X$ , в которой расположено выбранное число.
16. В массиве  $X(N)$  каждый элемент равен 0, 1 или 2. Переставить элементы массива так, чтобы сначала располагались все нули, затем все единицы и, наконец, все двойки (дополнительного массива не заводить).
17. Заданы число  $N$  и целочисленный массив  $A(N)$ . Найти длину самой длинной последовательности подряд идущих элементов массива, равных нулю.
18. Дан одномерный массив. Все его элементы, не равные нулю переписать (сохраняя их порядок) в начало массива, а нулевые элементы - в конец массива (новый массив не заводить).
19. Даны числа  $M, N$  и двумерный массив размером  $M \times N$ . Некоторый элемент этого массива назовем седловой точкой, если он является одновременно наименьшим в своей строке и наибольшим в своем столбце. Напечатать номера строки и столбца какой-нибудь седловой точки и напечатать число 0, если такой точки нет.
20. Даны числа  $N, K$  и два целочисленных массива  $X(N)$  и  $Y(K)$ . Можно ли в первом из них выбрать такие  $K$ , идущих подряд элементов, что  $X_{i+1} = Y_1, X_{i+2} = Y_2, \dots, X_{i+K} = Y_K$ ? Написать программу, которая решает эту задачу и печатает ответ ДА или НЕТ.
21. Слияние массивов. Даны два числа  $M, N$  и два упорядоченных массива  $a_1 \leq a_2 \leq \dots \leq a_m$  и  $b_1 \leq b_2 \leq \dots \leq b_n$ . Образовать из этих элементов упорядоченный массив  $c_1 \leq c_2 \leq \dots \leq c_{m+n}$ . Указание: обратить внимание на число действий программы при больших  $M$  и  $N$ .
22. Функция  $f(n)$  для целых неотрицательных  $n$  определена так:  
 $f(0) = 0, f(1) = 1,$   
 $f(2n) = f(n), f(2n+1) = f(n) + f(n+1).$   
Для данного  $N$  найти и напечатать  $f(N)$ . Обязательное условие:  $N$  столь велико, что недопустимо заводить массив из  $N$  чисел (равно как и массив, длина которого растет с ростом числа  $N$ ).
23. Напечатать в порядке возрастания все простые несократимые дроби, заключенные между 0 и 1, знаменатели которых не превышают 7.
24. Пусть  $P = (p_1, \dots, p_n)$  является перестановкой чисел  $1, 2, \dots, n$ . Таблицей инверсий перестановки  $P$  называется последовательность  $T = (t_1, \dots, t_n)$  в которой  $t_i$  равно числу элементов перестановки  $P$ , стоящих (в  $P$ ) левее числа  $i$  и больших числа  $i$ . Написать программу, которая по заданной таблице инверсий восстанавливает перестановку. Например:  $P = (5, 9, 1, 8, 2, 6, 4, 7, 3)$   $T = (2, 3, 6, 4, 0, 2, 2, 1, 0)$ .

25. Бит-реверс. Целое положительное число  $M$  записывается в двоичной системе счисления и разряды(в этой записи) переставляются в обратном порядке. Получившееся число принимается за значение функции  $B(M)$ . Напечатать значения  $B(M)$  для  $M=512, 513, 514, \dots, 1023$ .

26. Ввести число  $N$  заполнить двумерный массив размеров  $N \times N$  натуральными числами от 1 до  $N^2$  по спирали (рис).



27. Треугольник и точка. Заданы прямоугольные координаты  $x_1, y_1; x_2, y_2; x_3, y_3$  вершин треугольника и координаты  $x, y$  точки. Определить и напечатать, находится ли точка в треугольнике. Погрешностями вычислений пренебречь.
28. Квадратики. Дан массив  $A(M, M)$ , каждый элемент которого равен 0, 1, 5 или 11. Подсчитать в нем количество четверок  $A(i, j), A(i+1, j), A(i, j+1), A(i+1, j+1)$ , в каждой из которых все элементы различны.
29. Получить все расстановки 8 ладей на шахматной доске, при которых ни одна ладья не угрожает другой.
30. Получить все расстановки 8 ферзей на шахматной доске, при которых ни один ферзь не угрожает другому.
31. Имеется  $N$  предметов, веса которых равны  $A_1, A_2, \dots, A_n$ . Разделить эти предметы на две группы так, чтобы общий вес двух групп был максимально близким.
32. Лабиринт из  $N$  комнат задан таблицей соединений, в которой для каждой пары комнат указано, соединены ли они коридором. Построить путь из комнаты с номером  $I$  в комнату с номером  $J$ .
33. Лабиринт. Может ли путник выйти из лабиринта? Если может, то напечатать путь от выхода до начального положения путника. Лабиринт задан массивом  $A$  размером  $40 \times 40$ , в котором :  
 $A_{km}=0$ , если клетка  $(K, M)$  "проходимая";  
 $A_{km}=1$ , если клетка  $(K, M)$  "не проходимая".  
Начальное положение путника задается в проходимой клетке  $(I, J)$ . Путник может перемещаться из одной проходимой клетки в другую, если они имеют общую сторону. Путник выходит из лабиринта, когда попадает в граничную клетку (то есть в клетку  $(K, M)$ , где  $K$  или  $M$  равны 1 или 40).
34. Календарь. Заданы три числа  $A, B, C$ , которые обозначают число, месяц и год. Найти номер  $N$  этого дня с начала года. Указание: високосные годы - это те, у которых номер делится на 400, и те, у которых номер делится на 4, но не делится на 100.
35. Матрица содержит числа от 0 до 6. Покрыть матрицу костями домино из одного набора или сообщить, что это нельзя сделать.
36. Имеется  $n$  предметов с известным весом и стоимостью. Необходимо выбрать такие предметы, чтобы их суммарный вес был менее 30 кг, а стоимость – наибольшей. Напечатать суммарную стоимость выбранных предметов.
37. Ломаная задана координатами своих узлов. Сосчитать количество левых поворотов при движении вдоль ломаной.
38. Имеется бесконечная доска. За какое наименьшее число ходов конь сможет перейти с поля  $(x_1, y_1)$  на поле  $(x_2, y_2)$ ?
39. Есть множество предприятий, которые должны друг другу. Произвести взаимный зачет долгов на максимальную сумму.
40. Имеется два автобусных талона. Установить, пробиты ли они одним компостером или разными. Талон можно вкладывать в компостер любой стороной так, чтобы края талона были параллельны краям компостера и все отверстия компостера помещались на талоне. Отверстия компостера размещаются на квадратной сетке размером  $M \times N$ .
41. На квадратном клетчатом листе бумаги (размера  $100 \times 100$  клеток) нарисовано несколько прямоугольников. Каждый прямоугольник состоит из целых клеток. Различные прямоугольники не накладываются друг на друга и не соприкасаются. Задан массив размером  $100 \times 100$ , в котором элемент  $a_{ij}=1$ , если клетка  $(i, j)$  принадлежит какому-либо прямоугольнику, и  $a_{ij}=0$  в противном случае. Написать программу, которая сосчитает и напечатает число прямоугольников.

## Списки

1. Написать процедуру заполнения однонаправленного списка из  $N$  случайных чисел.
2. Написать процедуру заполнения двунаправленного списка из  $N$  случайных чисел.
3. Написать процедуру заполнения двоичного дерева из  $N$  случайных чисел.
4. Написать процедуру печати однонаправленного списка.
5. Написать процедуру печати двунаправленного списка.
6. Написать процедуру печати двоичного дерева.
7. Написать процедуру вставки числа в начало однонаправленного списка.
8. Написать процедуру вставки числа в начало двунаправленного списка.
9. Написать процедуру вставки числа в двоичное дерево.
10. Написать процедуру вставки числа в конец однонаправленного списка.
11. Написать процедуру вставки числа в конец двунаправленного списка.
12. Написать процедуру вставки числа в заданное место двунаправленного списка.
13. Написать процедуру вставки числа в заданное место однонаправленного списка.
14. Написать процедуру слияния двух двунаправленных списков.
15. Написать процедуру слияния двух однонаправленных списков.
16. Сохранить однонаправленный список в файл.
17. Сохранить двунаправленный список в файл.
18. Сохранить двоичное дерево в файл.
19. Задан файл, содержащий числа. Восстановить однонаправленный список из файла.
20. Задан файл, содержащий числа. Восстановить двунаправленный список из файла.
21. Задан файл, содержащий числа. Восстановить двоичное дерево из файла.
22. Написать процедуру удаления числа из заданного места двунаправленного списка.
23. Написать процедуру удаления числа из заданного места однонаправленного списка.
24. Написать процедуру удаления числа из двунаправленного списка.
25. Написать процедуру удаления числа из однонаправленного списка.
26. Отсортировать однонаправленный список.
27. Распечатать двунаправленный список в обратном порядке.
28. Сравнить два элемента двух однонаправленных списков.
29. Найти сумму элементов двоичного дерева.
30. Определить высоту двоичного дерева.
31. Напишите процедуру, определяющую пуст ли список.
32. Найти минимальный элемент двоичного дерева.
33. Найти минимальный элемент однонаправленного списка.
34. Найти минимальный элемент двунаправленного списка.
35. Найти максимальный элемент двоичного дерева.
36. Листом дерева называется такой узел, который не имеет предков. Подсчитать сколько листьев в данном дереве.
37. Напишите функцию с именем `Empty`, которая возвращает значение `true`, если данный список пуст. (Подсказка: передайте в функцию значение длины списка.)
38. Напишите функцию с именем `Full`, которая возвращает значение `true`, если в массиве, содержащем список, больше не осталось свободного места.
39. Напишите функцию с именем `Equal`, которая в качестве параметров получает два списка и возвращает значение `true`, если эти списки имеют одинаковую длину, а элементы в них попарно равны.
40. Напишите функцию с именем `Delete`, которая получает список и значение и ищет данное значение в списке. Если оно найдено, соответствующий элемент удаляется из списка, а следующие за ним элементы сдвигаются вверх, чтобы заполнить образовавшееся пустое место.
41. Напишите функцию с именем `DeleteAll`, которая удаляет из списка все элементы, значение которых равно `item`. При этом в списке не должно остаться пробелов.
42. Напишите функцию с именем `Component`, которая возвращает элемент списка, если заданный номер позиции (индексное значение) лежит в интервале от 0 до `length-1`. Функция также должна возвращать флаг (Булевского типа) с именем `valid`, значение которого равно `false`, если индекс выходит за границы указанного диапазона.

43. Напишите программу сравнения двух списков не предполагая, что списки являются равными по длине. Если один из списков длиннее другого, выведите сообщение о том, какой именно длиннее, и прекратите дальнейшее выполнение сравнения.
44. Напишите функцию, которая меняет местами максимальный и минимальный элементы непустого дерева.
45. Напишите функцию, которая заменяет в дереве все отрицательные элементы на их абсолютные величины.
46. Напишите функцию, которая определяет число вхождений элемента E в дерево.
47. Напишите функцию, которая определяет среднее арифметическое всех элементов непустого дерева.
48. Напишите функцию, которая подсчитывает число вершин на n-ом уровне непустого дерева. Корень считать вершиной нулевого уровня.
49. Напишите функцию, которая переворачивает список L, т.е. изменяет ссылки в этом списке так, чтобы его элементы оказались расположенными в обратном порядке.
50. Напишите функцию, которая в списке L из каждой группы подряд идущих равных элементов оставляет только один.
51. Написать функцию, которая проверяет на равенство два дерева.
52. Напишите функцию, которая определяет максимальную глубину непустого дерева, т.е. число ветвей в самом длинном из путей от корня дерева до листьев.
53. Написать функцию, которая в конец непустого списка L добавляет все его элементы, располагая их в обратном порядке (например: по списку 1 2 3 построить список 1 2 3 3 2 1).
54. Написать функцию, которая из списка L, содержащего не менее двух элементов, удаляет все элементы, у которых одинаковые «соседи» (первый и последний элементы считать соседями).

## Объекты.

### 1. Рациональное число.

Рациональное число - это число, которое может быть представлено в виде дроби с целочисленным числителем и знаменателем. Примерами рациональных чисел служат: 0.75 (то есть  $3/4$ ) и 1.125 (то есть  $9/8$ ). Значение **p** не является рациональным числом, если оно не может быть представлено в виде отношения двух целых чисел. Вашей задачей будет разработать, реализовать и проверить класс Rational, который должен представлять рациональное число в виде пары целых чисел, а не одного числа с плавающей точкой. Класс Rational должен иметь два конструктора. Первый из них должен позволять указывать начальные значения числителя и знаменателя. Второй (конструктор по умолчанию) создает рациональное число 0, представленное в виде числителя со значением 0 и знаменателя со значением 1. Вы должны обеспечить отработку следующих операций:

- конструкторы как по умолчанию, так и для явной инициализации объектов класса Rational
- арифметические операции сложения, вычитания, умножения и деления двух объектов класса Rational. Они должны быть реализованы в виде функций, возвращающих значение типа Rational
- Булевская операция сравнения (на равенство) двух объектов класса Rational
- операция вывода, печатающая значение типа Rational в форме числитель/знаменатель

Добавьте любые дополнительные операции, которые, по вашему мнению, могут быть полезными для работы с рациональными числами.

### 2. Комплексное число.

Комплексное («мнимое») число имеет вид **a+bi**, где **i** - это квадратный корень из -1. Здесь **a** называется действительной частью, а **b** - мнимой частью комплексного числа. Число **a+bi** также может быть представлено в виде упорядоченной пары вещественных чисел **(a,b)**.

Для двух комплексных чисел **(a,b)** и **(c,d)** справедливы следующие арифметические операции:

$$(a,b)+(c,d)=(a+b,c+d)$$

$$(a,b)-(c,d)=(a-b,c-d)$$

$$(a,b)*(c,d)=(a*c-b*d,a*d+b*c)$$

$$(a,b)/(c,d)=\left(\frac{a*c+b*d}{c^2+d^2}, \frac{b*c-a*d}{c^2+d^2}\right)$$

Кроме того, абсолютное значение (или модуль) комплексного числа определяется как

$$|(a,b)|=\sqrt{a^2+b^2}$$

Разработайте, реализуйте и протестируйте класс для хранения комплексных чисел, в котором действительная и мнимая части представляются как числа с двойной точностью (тип данных double). Класс должен предоставлять как минимум, следующий набор операций:

- конструкторы, как по умолчанию, так и для явной инициализации. Начальным значением по умолчанию должно быть (0.0, 0.0)
- арифметические операции сложения, вычитания, умножения и деления двух комплексных чисел. Они должны быть реализованы в виде функций, возвращающих объект класса.
- операция вычисления модуля комплексного числа
- две операции просмотра значений, - RealPart и ImagPart, - которые возвращают действительную и мнимую части комплексного числа.

### 3. Прямая.

Определите объект TLine - прямая. Прямая имеет координаты, толщину, цвет и может перемещаться в пространстве. Методами объекта являются:

- a). Init - устанавливает значения полей объекта и рисует прямую на экране методом Draw.
- b). Draw - рисует прямую, заданным цветом и толщиной в соответствующих координатах преобразованных к экранным координатам.
- c). Length - вычисляет длину прямой.
- d). Path - расстояние от прямой до другой прямой или плоскости.
- e). Parallel – определяет параллельность прямой к другой или плоскости.
- f). Side – определяет в какой полуплоскости относительно прямой находится точка.

### 4. Эллипс.

Определите объект TEllipse — эллипс. Эллипс определяется координатами двух точек (слева сверху

и справа внизу), толщиной и цветом линии, цветом заливки и может перемещаться в пространстве. Методами объекта являются:

- a). Init - устанавливает значения полей объекта и рисует эллипс на экране методом Draw.
- b). Draw - рисует эллипс, заданным цветом и толщиной в соответствующих координатах преобразованных к экранным координатам.
- c). Area - вычисляет площадь эллипса.
- d). Focus – определяет координаты фокусов эллипса и рисует их.
- e). Line – определяет и рисует прямые.
- f). Path – определяет минимальное расстояние от точки до эллипса.
- g). Compress – сжимает эллипс в выбранном направлении.
- h). Expand – расширяет эллипс в выбранном направлении.
- i). Fill – заливает внутренность эллипса, выбранным цветом.
- j). Inside – определяет, находится ли точка внутри эллипса или снаружи.

## 5. Окружность.

Определите объект TCircle — окружность. Окружность определяется координатами центра окружности, радиусом, толщиной и цветом линии, цветом заливки и может перемещаться в пространстве. Методами объекта являются:

- a). Init - устанавливает значения полей объекта и рисует окружность на экране методом Draw.
- b). Draw - рисует окружность, заданным цветом и толщиной в соответствующих координатах преобразованных к экранным координатам.
- c). Area - вычисляет площадь окружности.
- d). Curvature – определяет кривизну окружности.
- e). Line – определяет и рисует касательную прямую в точке, лежащей на окружности.
- f). Lines – определяет и рисует касательные прямые в точке, лежащей вне окружности.
- g). Path – определяет минимальное расстояние от точки до окружности.
- h). Compress – сжимает окружность.
- i). Expand – расширяет окружность.
- j). Fill – заливает внутренность окружности, выбранным цветом.
- k). Inside – определяет, находится ли точка внутри окружности или снаружи.

## 6. Шар.

Определите объект TSphere — шар. Шар определяется координатами центра шара, радиусом, толщиной и цветом линии, цветом заливки и может перемещаться и вращаться в пространстве. Методами объекта являются:

- a). Init - устанавливает значения полей объекта и рисует шар на экране методом Draw.
- b). Draw - рисует шар, заданным цветом и толщиной в соответствующих координатах преобразованных к экранным координатам.
- c). Area - вычисляет площадь поверхности шара.
- d). Volume – определяет объем шара.
- e). Segment – определяет объем сегмента шара. (Среза шара плоскостью.)
- f). Curvature – определяет кривизну шара.
- g). Line – определяет и рисует касательную плоскость в точке, лежащей на поверхности шара.
- h). Lines – определяет и рисует касательные прямые в точке, лежащей вне шара в выбранном направлении.
- i). Path – определяет минимальное расстояние от точки до шара.
- j). Compress – сжимает шар.
- k). Expand – расширяет шар.
- l). Fill – заливает внутренность шара, выбранным цветом.
- m). Inside – определяет, находится ли точка внутри шара или снаружи.
- n). Rotation – вращает шар с заданной скоростью в выбранном направлении.

## 7. Объект TFish.

Определите объект TFish — аквариумная рыбка. Рыбка имеет координаты, скорость, размер, цвет, направление движения. Методами объекта являются:

- a). Init — устанавливает значения полей объекта и рисует рыбу на экране методом Draw.
- b). Draw — рисует рыбу в виде уголка, с острием в точке Coord и направленного острием по ходу

движения рыбы.

- c). Look — проверяет несколько точек на линии движения рыбы. Если хоть одна из них отличается по цвету от воды, возвращается ее цвет и расстояние до рыбы.
- d). Run — перемещает рыбу в текущем направлении на расстояние, зависящее от текущей скорости рыбы.

## 8. Объект TBlock

Определите объект TBlock — блок игры тетрис. Он имеет координаты, тип блока, цвет. Методами объекта являются:

- a). Init — устанавливает значения полей объекта и рисует на экране методом Draw.
- b). Draw — рисует блок
- c). Look — проверяет можно ли падать вниз.
- d). Left, Right, Bottom — сдвиг блока в соответствующую сторону. Rotate — поворот блока.

## 9. Объект TShape

Определите объект TShape — фигура. Она имеет координаты, тип фигуры (круг, прямоугольник, овал, треугольник), размер, цвет, стиль закрашивания (S\olid\ - сплошное T\ransparent\ - контур, L\ine\ - в полосочку). Методами объекта являются:

- a). Init — устанавливает значения полей объекта и рисование на экране методом Draw.
- b). Draw — рисует фигуру.
- c). Delete — стирает фигуру
- d). Transform — изменяет произвольно фигуру в другой вид.

## 10. Объект TWindow

Определите объект TWindow — окно. Оно имеет координаты, размер, цвет, стиль (L — контур одинарной линией, DL — контур двойной линией). Методами объекта являются:

- a). Init — устанавливает значения полей объекта и рисует окно методом Draw.
- b). Draw — рисует окно.
- c). Close — стирает окно
- d). ReSize — изменяет размер окна в зависимости от передаваемого параметра.

## 11. Адрес.

Следующий класс представляет почтовый адрес в России.

```
type String20=string[20];
TAddress=object
public:
    procedure Write;
    Address( NewStreet; string20,      // in
            NewCity: string20,         // in
            NewState: string20,        // in
            NewZip: string20           // in
            );
private:
    Street:string20;
    City:string20;
    State: string20;
    ZipCode: string20;
end;
```

Используя наследование, мы хотим создать из Address класс для представления международного адреса - InterAddress. Предположим, что международный адрес имеет те же атрибуты, что и адрес в России, плюс код страны (строка, в которой указано название страны). Общедоступными операциями класса InterAddress являются: Write (которая замещает функцию Write, унаследованную от класса Address) и конструктор, который получает пять параметров (улица, город, область, почтовый индекс и код страны). Напишите объявление для класса InterAddress. Напишите реализацию для конструктора класса InterAddress. Напишите реализацию для функции-члена Write класса InterAddress. Напишите глобальную функцию PrintAddress, которая получает единственный параметр и использует динамическое связывание для вывода либо адреса в России, либо международного адреса. Внесите изменения в объявление класса Address, необходимые для корректной работы

функции PrintAddress.

## 12. Таймер.

Разработайте, реализуйте и протестируйте класс с именем Timer для таймера с обратным отсчетом. Этот класс имитирует настоящий таймер, отсчитывая секунды в обратном порядке, начиная с некоторого начального значения времени. Когда таймер достигает нуля, он должен запищать (это можно сделать, послав символ сигнала '\a' в стандартное устройство вывода). Для этого таймера нужно реализовать следующие операции:

- создать таймер, инициализировав его указанным количеством секунд
- запустить таймер
- переустановить таймер в некоторое значение

После запуска функции Start, она должна постоянно декрементировать и выводить текущее значение таймера примерно раз в секунду. Чтобы сделать в программе паузу на одну секунду, используйте цикл For, тело которого не выполняет никаких операций; то есть его телом является пустое выражение. Поэкспериментируйте с количеством итераций цикла, пока не достигнете паузы, как можно более близкой к одной секунде. Если в Вашей стандартной библиотеке есть функции для очистки экрана и позиционирования курсора в заданной точке экрана, вам может быть интересно сделать следующее. Сначала очистите экран. Затем всегда выводите значение таймера в одной и той же позиции в центре экрана. Каждое выводимое значение должно замещать собой предыдущее, как в настоящем таймере.

## 13. Неупорядоченный список.

Разработайте, реализуйте и протестируйте класс с именем IntList. Каждый объект этого класса является неупорядоченным списком не более чем из 100 значений. (Вспомните, что неупорядоченным называется список, элементы которого не отсортированы по значению.) Класс должен предоставлять как минимум следующий набор операций:

- создать пустой список
- проверить, является ли список пустым (true или false)
- проверить, является ли список полным (true или false)
- вставить в список указанное целое число
- удалить из списка указанное целое число
- осуществить поиск указанного целого числа, возвратив в зависимости от результата поиска true или false
- осуществить сортировку списка по возрастанию
- вывести все значения, содержащиеся в списке

Внимательно обдумайте предусловия каждой операции. Например, предусловие для операции Delete должно состоять в том, что список не является пустым. К счастью, клиент может проверить это предусловие, вызвав сначала операцию, которая сообщает, является ли список пустым. Вы также должны решить, позволено ли будет удалять из списка несуществующее целое число. Если да, то операция Delete не должна никак реагировать на это (и, вследствие постусловий, пользователь должен это понимать). Если же такая операция не будет разрешена, это также должно быть отражено в предусловиях. Тогда бремя проверок ошибок ложится на вызывающий код, а не на операцию Delete. Для конкретного представления данных для IntList вы должны использовать два компонента: массив из 100 элементов типа int и простая переменная для хранения текущей длины списка.

## 14. Упорядоченный список.

Создайте класс для упорядоченного списка с именем OrdIntList. Список должен всегда быть упорядоченным при операциях добавления и удаления элементов. Каждый объект этого класса является упорядоченным списком не более чем из 100 значений. (Вспомните, что упорядоченным называется список, элементы которого отсортированы по значению.) Класс должен предоставлять как минимум следующий набор операций:

- создать пустой список
- проверить, является ли список пустым (true или false)
- проверить, является ли список полным (true или false)
- вставить в список указанное целое число
- удалить из списка указанное целое число
- осуществить поиск указанного целого числа, возвратив в зависимости от результата поиска true



или false

g). вывести все значения, содержащиеся в списке

Внимательно обдумайте предусловия каждой операции. Например, предусловие для операции Delete должно состоять в том, что список не является пустым. К счастью, клиент может проверить это предусловие, вызвав сначала операцию, которая сообщает, является ли список пустым. Вы также должны решить, позволено ли будет удалять из списка несуществующее целое число. Если да, то операция Delete не должна никак реагировать на это (и, вследствие постусловий, пользователь должен это понимать). Если же такая операция не будет разрешена, это также должно быть отражено в предусловиях. Тогда бремя проверок ошибок ложится на вызывающий код, а не на операцию Delete. Для конкретного представления данных для OrdIntList вы должны использовать два компонента: массив из 100 элементов типа int и простая переменная для хранения текущей длины списка.

## 15. Прокат видеофильмов.

Ваши родители подумывают открыть пункт проката видеофильмов. Поскольку они хотят помочь вам в учебе, они попросили вас написать программу для обработки информации о продукции на складе. Какие основные объекты можно выделить в пункте проката? Вещи, которые берутся напрокат, и люди, которые их берут. Начните с абстракции вещей - видеокассет. Чтобы выяснить характеристики видеокассеты, составьте список вопросов:

- Должен ли объект представлять одну физическую кассету, или он может быть просто названием (чтобы иметь возможность хранить информацию о количестве копий)?
- Какую информацию следует хранить о каждом названии?
- Должен ли объект содержать номер карточки клиента, взявшего эту кассету напрокат?
- Если существует несколько копий кассеты, нужно ли хранить информацию о каждой копии?
- Какие операции должен выполнять объект, отражающий видеокассету? Допустим, вы решили, что объектом будет название, а не физический экземпляр кассеты. Количество копий этого фильма может быть членом объекта. В других членах будет содержаться название, имена актеров, исполняющих главные роли, имя продюсера, имя директора и название компании, создавшей фильм. Система должна следить за тем, кто какие кассеты взял; но это не является свойством самой кассеты. Потом вы решите, как представить отношение «взял напрокат». Сейчас пока не важно, у кого именно лежит каждая копия. У объекта, отражающего кассету, должны быть операции для его инициализации и обращения к различным членам, содержащим данные. В дополнение к этому, объект должен следить за количеством копий, определять, если ли копия в наличии, регистрировать выдачу и прием копий. Поскольку позже придется создавать список видеокассет, нужно включить такие операции для объекта, как сравнение и вывод заголовков.

Выполнить реализацию и проверку объекта и написать программу, которая делает следующее:

- Создает список из объектов, отражающих видеокассеты.
- Осуществляет поиск в списке указанного значения.
- Определяет, есть ли в пункте проката свободные копии указанного фильма.
- Выводит список названий видеофильмов.

Определите характеристики клиента и опишите объект, отражающий информацию о клиенте. Напишите соответствующие операции и протестируйте их. Используя свое представление клиента, напишите программу, которая выполняет следующие задачи:

- Создает список клиентов.
- Осуществляет поиск в списке по имени клиента.
- Осуществляет поиск в списке по идентификационному номеру клиента.
- Выводит список имен клиентов.

Соедините вместе списки видеокассет и клиентов в программе, обладающей следующими возможностями:

- Приемка видеокассет.
- Выдача видеокассет.
- Определить, сколько кассет взял указанный клиент (по идентификационному номеру клиента).
- Определить, у кого из клиентов находятся указанные фильмы (по названию фильма).

(Подсказка: создайте объект hasVideo, который содержит название фильма и номер клиента.)

## СТРУКТУРЫ ДАННЫХ.

### Структура данных 'стек'.

Стеком будем называть последовательность элементов, упорядоченных по времени их поступления. Эта последовательность доступна только с одного конца (вершины стека). Для работы со стеком необходим указатель вершины стека. Основные операции над стеком следующие: "запомнить в стеке" и "извлечь из стека" (причем извлекается последний из занесенных в стек элементов - то есть элемент с вершины стека). Поэтому говорят, что стек – это структура типа LIFO – "Last In - First Out" – "последним зашел – первым вышел". Для представления стека в программе обычно используется одномерный массив (назовем его В), нумерация элементов которого начинается с нуля. В этом нулевом элементе массива хранится индекс первого свободного места в массиве (т.е. увеличенный на 1 индекс вершины стека). Если массив пуст, то указатель равен 1 ( $B[0]=1$ ).

Добавление элемента X в стек реализуется очень просто – на первое свободное место (индекс которого хранится в  $B[0]$ ) помещается X, после чего индекс первого свободного места увеличивается на 1:

```
B[B[0]]:=x; { Занести в стек }  
B[0]:=B[0]+1; { Увеличить указатель }
```

Если необходимо извлечь элемент x из стека, то берется последний из занесенных элементов (естественно, только в том случае, если стек непуст), и указатель на первое свободное место уменьшается на 1:

```
if B[0]<>1 { Если стек не пуст }  
then  
begin  
x:=B[B[0]]; { Взять элемент }  
B[0]:=B[0]-1; { Уменьшить указатель }  
end;
```

### Структура данных 'список'.

Каждый элемент списка имеет указатель на следующий за ним элемент (другими словами – хранит информацию о расположении следующего элемента), а если список двусвязный (двунаправленный), то имеет также указатель и на предшествующий элемент. Кроме того, есть переменная, указывающая на первый элемент списка. Например, двунаправленный список может быть реализован с помощью двумерного массива  $A[1..3, 1..N]$ , где содержимое  $A[2, i]$  определяет позицию (индекс, адрес) элемента, предшествующего элементу  $A[1, i]$ , а содержимое  $A[3, i]$  определяет позицию элемента, следующего за  $A[1, i]$ .

Основные операции над списком следующие: "удалить из списка" и "поместить в список".

Например, помещение в список элемента x после элемента  $A[1, i]$  может выглядеть так:

```
A[1, j]:=x; { Занести в массив }  
A[2, j]:=i; { Изменить указатели }  
A[3, j]:=A[3, i];  
A[3, i]:=j;
```

Здесь j - адрес свободной позиции в массиве A.

### Структура данных 'очередь'.

Очередь в программировании, как и очередь в магазине, имеет начало и конец. Если приходит новый элемент, то он становится в конец очереди, если необходимо взять элемент из очереди, то он берется из ее начала. Очередь будем представлять в виде массива. Пусть у нас есть индекс первого - BEG и последнего - FIN элементов очереди (если очередь пуста, то  $BEG=FIN+1$ ; сначала же  $BEG=1$ ,  $FIN=0$ ).

Очередь опишем так: `var Queue=array[1..MaxQ] of element;`

Тут MaxQ - максимальное число элементов в очереди, element - какой-то тип данных. В качестве element можно взять структуру следующего вида:

```
type element = record
```

x: byte;

y: byte;

end;

где element - это запись, состоящая из x-овой и y-овой координаты.

С очередью можно проводить операции:

вставка в очередь InQueue,

удаление из очереди OutQueue.

Procedure InQueue (x : element);

begin

FIN:=FIN+1; { на первое свободное место }

Queue[FIN]:=x; { ставим элемент x }

end;

Procedure OutQueue (var x : element);

begin

x:=Queue[BEG]; { берем первый элемент }

BEG:=BEG+1; { и изменяем указатель }

{ на следующий за ним }

end;

## ЗАДАЧИ.

### Задача 1.

Вводится последовательность, состоящая из  $N$  пар символов  $(a_i, b_i)$ . Каждая пара определяет порядок предшествования символов, например, пара  $(b, c)$  означает, что символ "b" предшествует символу "c". Из порядка  $(b, c)$  и  $(c, a)$  следует порядок  $(b, a)$ . Необходимо определить, является ли введенная последовательность:

- а) полной, т.е. все использованные для формирования пар символы (выбросив повторяющиеся) можно выстроить в цепочку  $(A_1, A_2, \dots, A_s)$  в порядке предшествования;
- б) противоречивой, т.е. для некоторых символов  $x, y$  можно получить одновременно порядок как  $(x, y)$  так и  $(y, x)$ ;

### Задача 2.

Вокруг считающего стоит  $N$  человек, из которых выделен первый, а остальные занумерованы по часовой стрелке числами от 2 до  $N$ . Считающий, начиная с кого-то, ведет счет до  $M$ . Человек на котором остановился счет, выходит из круга. Счет продолжается со следующего человека и так до тех пор, пока не останется один человек.

Определить

- а) номер оставшегося человека, если известно  $M$  и то, что счет начинался с первого человека;
- б) номер человека с которого начинался счет, если известно  $M$  и номер оставшегося человека  $L$ .

### Задача 3.

Задана полоска длиной  $2^k$  клеток и шириной в одну клетку. Полоску сгибают пополам так, чтобы правая половинка оказалась под левой. Сгибание продолжают до тех пор, пока сверху находится больше одной клетки. Необходимо пронумеровать клетки таким образом, чтобы после окончания сгибания полосы номера клеток в получившейся колонке были расположены в порядке  $1, 2, 3, 4, \dots, 2^k$ .

### Задача 4.

Квадрат разбит на  $4^k$  равновеликих квадратных клеток. Квадрат перегибается поочередно относительно вертикальной (правая половина подкладывается под левую) и горизонтальной (нижняя половина подкладывается под верхнюю) оси симметрии до тех пор, пока все клетки не будут расположены друг под другом. Требуется занумеровать клетки исходного квадрата таким образом, чтобы в результате выполнения операций перегиба номера клеток, расположенных друг под другом, образовали числовую последовательность  $1, 2, 3, \dots, 4^k$ , начиная с верхней клетки.

### Задача 5.

Даны обозначения двух полей шахматной доски (например,  $A5$  и  $C2$ ). Найти минимальное число ходов, которые нужны шахматному коню для перехода с первого поля на второе.

### Задача 6.

Дана конечная последовательность, состоящая из левых и правых скобок различных заданных типов. Как определить, можно ли добавить в нее цифры и знаки арифметических действий так, чтобы получилось правильное арифметическое выражение.

### Задача 7.

В таблице  $A$  размера  $N$  за один просмотр необходимо каждый элемент заменить на ближайший следующий за ним элемент, который больше его. Если такого элемента нет, то заменить его на ноль. Можно использовать дополнительную память.

ПРИМЕР  $A = 1\ 3\ 2\ 5\ 3\ 4$

ОТВЕТ  $A = 3\ 5\ 5\ 0\ 4\ 0$

### Задача 8.

В городе расположены  $n$  автобусных остановок, обозначенных числами из  $N=\{1,2,\dots,n\}$ . Имеется  $R$  автобусных маршрутов, заданных последовательностями соседних остановок при движении автобуса в одном направлении:

$M1=\{I11,I12,\dots,I1m1\}$ ,

$M2=\{I21,I22,\dots,I2m2\}$ ,

.....

$Mr=\{Ir1,Ir2,\dots,Irmr\}$ , где  $I_{jk}$  натуральное.

Написать программу, которая по заданным номерам остановок  $I$  и  $J$  определяет наиболее быстрый путь перемещения пассажира из остановки  $I$  в остановку  $J$  с использованием имеющихся маршрутов автобусов при условии, что время движения между соседними остановками у всех маршрутов одинаково и в 3 раза меньше времени изменения маршрута. Кроме того, автобусы могут двигаться в обоих направлениях.

### Задача 9.

Одинокый король долго ходил по бесконечной шахматной доске. Известна последовательность из  $N$  его ходов (вверх, вниз, влево, вправо, вверх-влево и т.п.). Написать программу, определяющую побывал ли король дважды на одном и том же поле за минимально возможное при заданном  $N$  число вычислений.

### Задача 10.

По кругу расположено  $N$  монет гербами вверх и  $M$  монет гербами вниз. Обходя круг по ходу часовой стрелки, переворачивает каждую  $S$ -тую монету. В первый раз счет начинается с герба. В каком порядке надо расставить монеты, чтобы после  $K$  ходов стало  $L$  монет, лежащих гербами вверх.

### Задача 11.

$N$  серых и  $M$  белых мышей сидят по кругу. Кошка ходит по кругу по часовой стрелке и съедает каждую  $S$ -тую мышку. В первый раз счет начинается с серой мышки. Составить алгоритм определяющий порядок в котором сидели мышки, если через некоторое время осталось  $K$  серых и  $L$  белых мышей.

### Задача 12.

Из листа клетчатой бумаги размером  $M \times N$  клеток удалили некоторые клетки. На сколько кусков распадется оставшаяся часть листа?

Пример. Если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.

МОДИФИКАЦИЯ. То же, но перед удалением клеток лист склеили в цилиндр высотой  $N$ .

### Задача 13.

Имеется  $n$  черных и белых карточек, сложенных в стопку. Карточки раскладываются на стол в одну линию следующим образом: первая кладется на стол, вторая под низ стопки, третья-на стол, четвертая – под низ стопки и т.д., пока все карточки не будут выложены на стол. Каким должно быть исходное расположение карточек в стопке, чтобы разложенные на столе карточки чередовались по цвету: белая, черная, белая, черная и т.д.

### Задача 14.

Форма тела задана матрицей  $A$  размерности  $M \times N$ . Элементы матрицы – натуральные числа. Элемент  $A(i,j)$  соответствует высоте горизонтальной квадратной площадки размера  $1 \times 1$  относительно нижнего основания. Нижнее основание формы горизонтально.

Определить объем невытекшей воды, если

а) Тело опускается полностью в воду, затем поднимается;

Пример:

5 3 1

$A = 5 \ 2 \ 5$

2 5 5

После подъема вода останется только во внутреннем углублении, над элементом  $A(2,2)=1$ .

Объем воды равен 1.

б) В позицию  $(i_0, j_0)$  выливается объем воды  $V$ .

### Задача 15.

Матрица размера  $N \times M$  определяет некоторый лабиринт. В матрице элемент 1 обозначает стену, а 0 определяет свободное место. В первой строке матрицы определяются входы  $x(i)$ , а в последней выходы  $y(i)$ ,  $i=1, \dots, k$ , которые должны быть нулевыми элементами.

Необходимо определить, можно ли

а) провести  $k$  человек от входа  $x(i)$  до выхода  $y(i)$  соответственно,  $i=1, \dots, k$ , таким образом, чтобы каждое свободное место посещалось не более одного раза;

б) то же, но человека можно выводить через любой из выходов.

Примечание: Движение в лабиринте осуществляется только по вертикали или горизонтали.

### Задача 16.

Есть министерство из  $N$  чиновников, где  $N$  натуральное число. У каждого из чиновников могут быть как подчиненные, так и начальники, причем справедливы правила: подчиненные моего подчиненного – мои подчиненные, начальники моего начальника – мои начальники, мой начальник не есть мой подчиненный, у каждого чиновника не более одного непосредственного начальника.

Для того чтобы получить лицензию на вывоз меди необходимо получить подпись 1-ого чиновника – начальника всех чиновников. Проблема осложняется тем, что каждый чиновник, вообще говоря, может потребовать "визы", т.е. подписи некоторых своих непосредственных подчиненных и взятку – известное количество долларов. Для каждого чиновника известен непустой список возможных наборов "виз" и соответствующая каждому набору взятка. Пустой набор означает, что данный чиновник не требует виз в данном случае. Чиновник ставит свою подпись лишь после того, как ему представлены все подписи одного из наборов "виз" и уплачена соответствующая взятка.

Необходимо определить и вывести минимальный по сумме уплаченных взяток допустимый порядок получения подписей для лицензии и стоимость.

$N < 100$ . Количество наборов для каждого чиновника не превосходит 15.

## РЕШЕНИЕ ЗАДАЧ ИЗ РАЗДЕЛА “СТРУКТУРЫ ДАННЫХ”.

### Задача 1.

Пусть при записи этих  $N$  пар встретилось всего  $K$  различных символов, которые образуют множество  $X$ . Идея решения задачи состоит в последовательном присвоении каждому символу  $s$  из  $X$  номера, который определяет количество  $P(s)$  элементов, предшествующих ему, с учетом свойства транзитивности (из  $(c,b)$  и  $(b,a)$  следует  $(c,a)$ ). Это осуществляется следующим образом:

Первоначально предполагается, что каждому символу не предшествует ни один символ, т.е.  $P(s)=0$  для любого  $s$ .

При просмотре очередной пары  $(x,y)$  символу  $y$  присваивается большее из значений  $P(x)+1, P(y)$ .

Очевидно, что при первом просмотре всех пар из входной последовательности определятся все упорядоченные цепочки длины 2, т.е. состоящие из 2 элементов. Поэтому номера некоторых элементов будут как минимум 1. При каждом из следующих просмотров входной строки возможно несколько вариантов.

1. Не произошло изменения ни одного из номеров символов. Если при этом номера символов различны и лежат в пределах от 0 до  $N-1$ , то эта нумерация определяет полный порядок. Иначе порядок неполный.

2. Номер некоторого символа превысил  $N-1$ . Это произошло потому, что рост номеров неограничен, т.е. осуществляется циклически. Следовательно порядок противоречив.

Легко понять, что число просмотров не превысит  $N$ .

### Вариант 2.

Рассмотрим следующий метод: Заведем массивы  $A$ : array  $[1..N, 0..N]$  of byte и

$Cnt$ : array  $[1..N]$  of byte; сначала  $A[i,0]=0$  и  $Cnt[i]=0$  для любого  $i$ .

Пусть среди  $2*N$  символов, образующих  $N$  пар, есть ровно  $K$  различных. Перенумеруем их от 1 до  $K$ . Будем считать, что пары составлены не из символов, а из соответствующих им номеров.

В  $i$ -ю строчку матрицы  $A$  будем заносить те элементы, которые являются вторыми элементами в парах с первым элементом  $i$ . В  $A[i,0]$  будет храниться текущее число этих элементов. Обработка пары  $(i,j)$  будет выглядеть следующим образом:

$A[i,0]:=A[i,0]+1$ ; { количество увеличилось на 1 }

$A[i,A[i,0]]:=j$ ; { вставляем  $j$  на первое свободное место }

В  $Cnt[i]$  будет храниться число пар, у которых элемент  $i$  является вторым в паре.

Если все символы без повторений, использованные для записи пар, можно выписать в цепочку в порядке предшествования, то у этой цепочки должен быть первый символ  $s$ , у которого нет предшествующего и которому соответствует  $Cnt[s]=0$ .

Может быть несколько ситуаций:

1. Такой элемент единственный – следовательно, это начало цепочки. Отбрасываем  $s$  из цепочки и убираем все пары с первым элементом  $s$  из множества пар, корректируя при этом массив  $Cnt$ :

for  $i:=1$  to  $A[0,s]$  do

$Cnt[A[s,i]]:=Cnt[A[s,i]]-1$ ;

после чего опять ищем элемент  $s$ , у которого нет предшествующего и которому соответствует  $Cnt[s]=0$ .

2. Таких элементов несколько, следовательно, между ними нельзя определить порядок предшествования – система неполна.

3. Таких элементов нет – следовательно, система противоречива.

### Задача 2.

Для решения этой задачи используем структуру данных 'список'. Каждый элемент списка указывает на следующий за ним элемент (другими словами – хранит информацию о расположении следующего элемента).

а). Заведем массив  $A$  из  $N$  ячеек - по числу людей в круге. В каждую из ячеек  $A[i]$  занесем номер стоящего следующим за  $i$ -ым человека. Первоначально  $A[i]=i+1, i=1,...,N-1, A[N]=1$ . Начиная счет от текущего человека (человека с номером  $IndTek$ , с самого начала  $IndTek=1$ ) будем делать ход – отсчитывать  $M$  ячеек, двигаясь по кругу в поисках человека, который должен выйти из круга. С учетом определения массива  $A$

это будет выглядеть следующим образом:

```
{ IndTek – это номер человека, с которого начинается счет }
for i:=1 to M-1 do { Отсчитываем M человек, начиная с IndTek }
begin
IndPred:=IndTek; { в IndPred сохраняем номер текущего
человека в круге }
IndTek:=A[IndTek]; { и вычисляем номер следующего за ним }
end;
```

После выполнения этого цикла в переменной IndTek будет номер человека, на котором остановился счет (т.е. номер человека, который покинул круг), а в переменной IndPred – номер предшествующего ему человека в круге.

Удалим человека с номером IndTek. Для этого мы просто изменим ссылку A[IndPred] у предшествующего ему человека так, чтобы он указывал не на IndTek, а на следующего за IndTek, т.е. на A[IndTek], и новый отсчет M-того человека начнем со следующего за удаленным:

```
A[IndPred]:=A[IndTek];
IndTek:=A[IndTek]; { Новый номер начального человека }
```

Все вышеописанные операции мы будем повторять до тех пор, пока в круге не останется одного единственного человека, т.е. до тех пор, пока A[IndTek] не станет равным IndTek, что означает, что следующим за человеком с номером IndTek является он сам.

Полностью весь фрагмент программы будет выглядеть так:

```
{ Инициализация массива A }
for i:=1 to N-1 do
A[i]:=i+1;
A[N]:=1;
{ IndTek – это номер человека, с которого начинается счет }
IndTek:=1;
while A[IndTek] <> IndTek do
begin
for i:=1 to M-1 do { Отсчитываем M человек, начиная с IndTek }
begin
IndPred:=IndTek; { в IndPred сохраняем номер текущего
человека в круге }
IndTek:=A[IndTek]; { и вычисляем номер следующего за ним }
end;
A[IndPred]:=A[IndTek];
IndTek:=A[IndTek]; { Новый номер начального человека }
end;
writeln('Номер последнего оставшегося человека ',IndTek);
```

Решения пункта b).

Будем считать, что человек с номером  $N+i$  – это то же самое, что и человек с номером  $i$ .

Предположим, что как и в пункте а), что мы начали счет с первого человека, выбрасывали из круга M-того, и последний оставшийся в круге человек имеет номер K. Очевидно, что если бы мы начали счет со второго человека, то последний оставшийся в круге человек имел бы номер  $K+1$ , ..., если с j-го, то  $K+j-1$ .

Если номер оставшегося человека L, то из равенства  $L=K+j-1$  определяем номер j первого человека (если  $j \leq 0$ , то заменим j на  $j+N$ , считая как и раньше, что человек с номером  $N+j$  – это то же самое, что и человек с номером j).

### Задача 3.

Будем моделировать сложение полосы, затем пронумеруем получившуюся колонку клеток числами от 1 до  $2^n$ , после чего распечатаем числа, приписанные первой, второй, ...,  $2^n$ -ой клетке исходной полосы.

Сначала создаем двусвязный список, состоящий из  $2^k$  элементов. Поле next будет указывать на элемент



находящийся под данным, а поле  $last$  - на элемент находящийся над данным. Для верхнего элемента  $last=0$ , а для нижнего  $n_{ext}=n+1$ , где  $n$ -общее число элементов. Вначале длина верхней полоски равняется  $n$  элементов, после первого сгибания их она станет  $n/2$ , после второго --  $n/4$ , и т.д. Пусть в данный момент длина верхней полоски есть  $sp$  элементов. Значит нам необходимо  $sp/2$  правых элементов опустить под  $sp/2$  левых. Для этого запустим цикл, который будет менять  $i$  от 1 до  $sp/2$  и на каждом шаге помещать  $(sp-i+1)$ -ю колонку элементов под  $i$ -ю, при этом порядок элементов в  $(sp-i+1)$ -ой колонке меняется на противоположный. После каждого сгибания  $sp$  уменьшается вдвое. Так продолжаем до тех пор пока  $sp>1$ . Попробуйте найти формулу и написать программу, которая без моделирования складывания полоски по номеру клетки выдавала бы ее номер.

#### Задача 4.

Решение этой задачи аналогично решению предыдущей задачи со сгибанием полоски с той лишь разницей, что тут будет поочередно производится два сгибания: сначала правая половина под левую, а затем нижнюю под верхнюю

#### Задача 5.

Идея решения основывается на использовании очереди. Вначале в очередь помещается элемент, определяющий исходное положение шахматного коня, а соответствующая клетка поля помечается как посещенная.

На каждом из следующих шагов алгоритма (пока очередь не пуста или не помечена конечная клетка) выполняются следующие действия.

1. Из очереди извлекается очередной элемент, который определяет некоторую позицию  $(x,y)$ .
2. Находятся клетки в пределах поля, которые достижимы из  $(x,y)$  одним ходом шахматного коня, и которые еще не помечены.
3. Элементы, соответствующие найденным клеткам, помещаются в очередь, а клетки помечаются как посещенные.

При этом вначале все клетки должны быть непомечены, а метку желательно ставить такую, чтобы можно было восстановить путь между полями.

Для этого метку поля можно определить как позицию поля, из которой она была помечена.

#### Задача 6.

Идея решения основывается на использовании стека. Считывая входную последовательность скобок за скобкой, выполняются следующие действия.

1. Если очередная скобка - открывающая, то помещаем ее в стек.
2. Если очередная скобка - закрывающая, то анализируем скобку, стоящую в вершине стека. Возможно несколько ситуаций:
  - а) открывающая скобка соответствует очередной закрывающей. В этом случае она выталкивается из стека, а процесс с новой входной скобкой.
  - б) открывающая скобка не соответствует очередной закрывающей или стек пуст. В этом случае невозможно получить правильное арифметическое выражение.

Когда все скобки входной строки обработаны, возможны 2 ситуации.

1. Стек пуст. В этом случае можно получить правильное арифметическое выражение.
2. Стек не пуст. В этом случае невозможно получить правильное арифметическое выражение.

#### Задача 7.

Необходимо организовать стек для позиций элементов, которые претендуют быть большими. Для каждого текущего элемента выталкивать из стека все позиции, на которых стоят элементы меньше текущего и заменить их текущим. Затем позицию текущего элемента поместить в стек. После просмотра всех элементов в стеке будут стоять позиции элементов, которые надо заменить на ноль.

#### Задача 8.

Идея решения основывается на использовании очередей. Но если реализация движения от остановки к остановке по одному маршруту аналогична технике, используемой в задаче 6, то возможность изменения маршрутов несколько усложняет структуру элемента в очереди.

Для этого для каждой остановки введем тройку чисел  $(O, M, Z_a)$ , где  $O$  – номер остановки,  $M$  – номер маршрута,  $Z_a$  – величина задержки. При выборе очередного элемента из очереди возможны 2 ситуации.

1. Продолжается движение по тому же маршруту. В этом случае в очередь заносятся тройки типа  $(O', M, Z_a)$ , где  $O'$  – номер остановки, соседней  $O$ , в маршруте  $M$ , а  $Z_a=0$ .

2. Изменяется маршрут. В этом случае в очередь заносятся тройки типа  $(O, M', Z_a)$ , где  $M'$  – номер измененного маршрута,  $Z_a=3$  (задержка на пересадку).

При этом новые тройки порождаются только тройками с задержкой, равной 0. В случае тройки с ненулевой задержкой она вновь помещается в очередь с уменьшенным на 1 значением задержки.

### Задача 9.

Будем хранить в массиве  $A[1..N, 1..2]$  координаты тех клеток, где король уже был. Возьмем  $N$  достаточно большим,  $N=500$ .  $A[i, 1]$  – икс-координата,  $A[i, 2]$  – игрековая координата клетки. Пусть в переменной  $DLN$  хранится количество уже сделанных ходов (т.е. в массиве  $A$  после хода номер  $i$  будут заняты клетки с первой по  $i$ -ую).

Результаты очередного хода заносим в массив  $A$  на первое свободное место (т.е. на место с индексом  $DLN+1$ ), увеличиваем количество сделанных ходов  $DLN:=DLN+1$ , и проверяем, совпадает ли хоть одна из позиций, на которых король уже был, с текущей позицией:

for  $i:=1$  to  $DLN-1$  do

if  $(A[i, 1]=A[DLN, 1])$  and

$(A[i, 2]=A[DLN, 2])$  { проверка совпадения }

then begin

writeln('совпадение на ходах ',  $DLN$ , ' и ',  $i$ );

halt; { останов }

end;

Давайте посмотрим, как можно вводить ходы короля. Он из клетки  $k$  может пойти на 8 соседних позиций.

1 2 3 Будем вводить направление перемещения короля

8 K 4 цифрой от 1 до 8.

7 6 5

Заведем массив  $XOD=array[1..8, 1..2]=((-1, 1), (0, 1), (1, 1), (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0))$ ;

В массив заносятся смещения по  $x$ - и  $y$ -координатам, соответствующие направлениям от 1 до 8. Координаты короля после сделанного хода вычисляются по формулам

$x:=x+XOD[i, 1]$ ;  $y:=y+XOD[i, 2]$ ;

### Задача 10.

Монеты лежат на  $N+M$  позициях. Пронумеруем эти позиции по порядку по контуру от 1 до  $N+M$ .

Заведем массив  $A$  из  $N+M$  ячеек. Первоначально все ячейки нулевые. Начиная счет от первой ячейки, будем делать ход – отсчитывать  $S$  ячеек (считаем, что за  $N+M$ -ым элементом следует непосредственно 1-ый элемент массива) и заменять в этой ячейке число  $i$  на число  $1-i$  (т.е. 0 на 1, а 1 на 0). После  $k$ -того хода остановимся.

Рассмотрим возникшую ситуацию. После каждого хода переворачивается одна монета, при этом разность количества монет, лежащих гербами вверх и количества монет, лежащих гербами вниз либо увеличивается, либо уменьшается на 2. Например, если переворачивается монета, лежащая гербом вверх, то при этом увеличивается на 1 количество монет гербом вниз и уменьшается на 1 количество монет гербом вверх.

Предположим, что после  $k$  ходов в массиве  $A$  стало  $p$  единиц – т.е.  $p$  монет, по сравнению с начальным положением, будут перевернуты после  $k$ -ого хода.

В случае, если  $L \geq N$ , то  $(L-N)$  монет, которые лежали гербами вниз, должны после  $k$ -ого хода быть перевернуты гербами вверх. (Если  $p < (L-N)$ , то это, очевидно, невозможно сделать). Среди оставшихся  $p-(L-N)$  перевернутых монет должна быть половина гербами вверх и половина – вниз, чтобы при перевороте суммарное число монет гербами вверх и вниз не изменилось. Следовательно, число  $p-(L-N)$  должно быть

четным, иначе условию задачи удовлетворить нельзя. Пусть  $p-(L-N) = 2v$ . Должно быть, очевидно,  $v \leq N$ ,  $v+(L-N) \leq M$ .

Итак, в случае  $L \geq N$ , если не выполняется хотя бы одно из неравенств

$$p-(L-N)=2v \geq 0,$$

$$v \leq N,$$

$v+(L-N) \leq M$ , то преобразование начальной конфигурации в конечную невозможно.

Иначе на  $(L-N)$  мест, помеченных в массиве  $A$  единицами, выставляем монеты гербами вниз. На оставшиеся  $2v=p-(L-N)$  помеченных единицами позиций кладем  $v$  монет гербами вниз и  $v$  гербами вверх в произвольном порядке. На остальные позиции кладем оставшиеся монеты опять же в произвольном порядке, чтобы в общей сложности было  $N$  монет гербами вверх и  $M$  – гербами вниз.

Но мы еще не учли тот факт, что счет начинается с первой монеты гербом вверх:

а) Если в массиве  $A$  первый элемент нулевой, то в случае, если среди  $(N+M)$ -р монет есть хоть одна гербом вверх (что эквивалентно выполнению условия  $N-v > 0$ ), то ее и кладем в первую позицию. Если все  $(N+M)$ -р монеты – гербом вниз (т.е.  $N-v=0$ ), то размещение монет невозможно.

б) Если же  $A[1]=1$ , то среди  $p$  монет должна быть по крайней мере одна, которую необходимо положить гербом вверх, иначе, опять же, размещение невозможно.

Случай  $N-L > 0$  разбирается аналогично.

### Задача 11.

Как и в задаче 10 заполняем сначала массив  $A$  нулями, перенумеровываем по порядку  $N+M$  позиций от 1 до  $N+M$ . Начиная с первой позиции (серая мышка) делаем ход – отсчитываем  $S$  нулевых позиций по порядку (считаем, что позиция, где сидела съеденная мышка, помечается единицей, несъеденная мышка – нулем; за  $N+M$ -ой позицией располагается первая) и выставляем в соответствующую позицию 1 – мышка съедена. Далее отсчет начинаем со следующей за съеденной мыши. (Для более быстрого поиска  $S$ -той мышки среди оставшихся в круге можно использовать список, описанный в задаче 2).

Делаем  $P=(N+M)-(K+L)$  ходов.

По условию задачи в первой позиции сидит серая мышка. Есть  $A[1]=1$  (первая мышь была съедена), то в оставшихся  $P-1$  единичной позиции в произвольном порядке расставляем  $N-K-1$  серых и  $ML$  белых мышей. В оставшихся незанятыми позициях рассаживаем опять же в произвольном порядке оставшихся мышей.

Если  $A[1]=0$ , и  $K=0$ , то начальной расстановки не существует (все серые мыши съедены, а должна остаться еще одна в первой позиции); если же  $K > 0$ , то в единичных позициях рассаживаем  $N-K$  серых и  $ML$  белых мышей, а в оставшихся позициях – в первую позицию – серую мышь, а во все остальные – белых и серых в произвольном порядке.

### Задача 12.

Идея решения состоит в последовательном определении множества клеток, принадлежащих тому же куску, что и некоторая фиксированная клетка.

Начиная с некоторой непомеченной клетки (начальной), которая помечается, помечаются все клетки, имеющие с ней общую сторону. Затем, используя вновь помеченные клетки, помечается множество клеток, имеющих с ними общую сторону и т.д., пока на некоторой итерации не будет помечено ни одной новой клетки. Множество клеток, помеченных таким образом, и составляет единый кусок листа максимального размера.

Для подсчета общего количества кусков необходимо подсчитать количество клеток, выбранных в качестве начальных. Новая начальная клетка выбирается из множества клеток, не помеченных на предыдущих итерациях. Для хранения вновь помеченных клеток можно использовать очередь или стек, а пометку клеток можно проводить в матрице размера  $N \times M$ .

### Задача 13.

Проще всего взять и промоделировать раскладку карточек: берем стопку из  $n$  черных и белых карточек и начинаем расклад как говорится в задаче – первая на стол, вторая под низ, третья на стол, и т.д. при этом первой выложенной карточке приписывается белый цвет, а каждой последующей – цвет, не совпадающий с

цветом предыдущей карточки. Повторяет так, пока не найдем раскраску всех  $n$  карточек.

Для этой задачи возьмем такую структуру данных как список. У списка есть начало и конец, и каждый элемент его указывает на следующий за ним элемент. Будем считать, что у последнего элемента списка указатель на следующий за ним равен 0. Список можно представлять массивом. Например, если в списке 4 элемента, то массив будет выглядеть так:

	1	2	3	4
A	2	3	4	5

т.е. за первым элементом находится  $A[1]=2$  элемент, ..., за 4-ым -  $A[4]=0$  (т.к. 4-ый элемент списка последний).

Пусть у нас есть указатель на начальный элемент списка и на последний (BEG и FIN соответственно). В списке  $n$  элементов.

Рассмотрим процедуру удаления элемента из начала списка (это соответствует переносу карточки на стол):

$BEG:=A[BEG]$ ; {теперь новый первый элемент списка - второй элемент старого списка}

Рассмотрим операторы перестановки элемента из начала списка в конец (это соответствует перемещению карточки сверху стопки под низ ее):

$A[FIN]:=BEG$ ; {следующей за последним элементом - бывший первый}

$FIN:=BEG$ ; {меняем ссылку на последний элемент}

$BEG:=A[BEG]$  {новый первый элемент}

$A[FIN]:=0$  {корректировка ссылки у последнего элемента}

Фрагмент программы будет выглядеть так:

for  $i:=1$  to  $N-1$  do  $A[i]:=i+1$ ;

$A[N]:=0$ ; {установка ссылок в списке}

$BEG:=1$ ;  $FIN:=N$ ;

$COLOR:=1$ ; {белый цвет = 1, черный = 0}

while  $A[BEG] \neq 0$  do {пока первый элемент не является}

{одновременно и последним}

begin

$BEFORE:=BEG$ ; {сохраняем индекс начала списка}

$BEG:=A[BEG]$ ; {удаляем первый элемент из списка}

$A[BEFORE]:=COLOR$ ; {раскрашиваем удаленный элемент}

{в нужный цвет}

$COLOR:=1-COLOR$ ; {меняем цвет}

$A[FIN]:=BEG$ ; {переставляем элемент из}

$FIN:=BEG$ ; {начала списка в конец}

$BEG:=A[BEG]$ ;

$A[FIN]:=0$

end;

$A[BEG]:=COLOR$ ; {раскрашиваем последний элемент}

{списка}

for  $i:=1$  to  $N$  do {распечатка цветов}

if  $A[i]=0$

then writeln('элемент',  $i$ , ' - черный')

else writeln('элемент',  $i$ , ' - белый');

#### Задача 14.

Для реализации данного алгоритма нам понадобится структура данных "очередь". Очередь в программировании, как и очередь в магазине, имеет начало и конец. Если приходит новый элемент, то он становится в конец очереди, если необходимо взять элемент из очереди, то он берется из ее начала. Очередь будем представлять в виде массива. Пусть у нас есть индекс первого - BEG и последнего - FIN элементов очереди (если очередь пуста, то  $BEG=FIN+1$ ; сначала же  $BEG=1$ ,  $FIN=0$ ).

Очередь опишем так: `var Queue=array[1..MaxQ] of element;`

Тут  $MaxQ$  - максимальное число элементов в очереди, `element` - какой-то тип данных. В задаче ниже в

качестве element можно взять такой

type element = record

x: byte;

y: byte;

end;

где element - это запись, состоящая из x-овой и y-овой координаты элемента матрицы  $A = \text{array}[0..M+1, 0..N+1]$  of integer. Индексы матрицы принимают такое значение из-за того, что мы будем окаймлять матрицу нулевыми элементами (так будет проще проверять выливание за край фигуры).

С очередью можно проводить операции:

вставка в очередь InQueue,

удаление из очереди OutQueue.

Procedure InQueue (x : element);

begin

FIN:=FIN+1; { на первое свободное место }

Queue[FIN]:=x; { ставим элемент x }

end;

Procedure OutQueue (var x : element);

begin

x:=Queue[BEG]; { берем первый элемент }

BEG:=BEG+1; { и изменяем указатель }

{ на следующий за ним }

end;

Находим максимальный элемент D в матрице A. Пока все элементы в матрице A не станут равны D, повторяем следующую последовательность действий:

а) Находим в матрице минимальный ненулевой элемент z (если их несколько, то берем любой из них), и заносим его в очередь (очередь сначала пустая). Если этот минимальный ненулевой элемент  $z=D$ , то СТОП. Сначала считаем, что  $p=D$  – это минимальный граничный элемент области, заполненной элементами со значением z.

б) Для каждого еще не просмотренного элемента очереди (пусть в матрице этот элемент находится в позиции (i,j)) повторяем следующее:

Для каждого соседа (сверху, снизу, справа, слева) данного элемента (i,j) проводим проверку-просмотр:

ЕСЛИ (сосед  $\leq$  z)

ТО  $P = \min\{P, \text{сосед}\}$

ИНАЧЕ ЕСЛИ сосед еще не просмотрен

ТО координата соседа - в очередь (в очереди появился еще один непросмотренный элемент) т.е. мы ищем минимальный окаймляющий элемент области, заполненной элементами со значением z.

Фрагмент программы поиска:

var Delta = array [1..4, 1..2] of integer =

((0,1),(1,0),(-1,0),(0,-1)); { Delta - возможные смещения соседних клеток от текущей клетки }

Current, neighbor : element;

z : integer;

....

{ Будем обозначать то, что элемент в матрице уже просмотрен }

{ умножением его на -1 }

{ минимальный ненулевой элемент матрицы имеет значение z }

while BEG  $\leq$  FIN+1 do begin

OutQueue(Current);

for i:=1 to 4 do begin

neighbor.x:=Current.x+Delta[i,1],

```

neighbor.y:=Current.y+Delta[i,2],
if A[neighbor.x,neighbor.y]=z
then InQueue(neighbor)
else p:=min(A[neighbor.x,neighbor.y],p);
end;
end;

```

Если в очереди нет больше непросмотренных элементов, то, если  $p < z$  (случай, когда данная область имеет "слив" за границы матрицы) в матрице во все просмотренные элементы из очереди заносим значение D (делаем их равными накопленному значению, чтобы больше их не рассматривать).

Если же  $p > z$ , то высчитываем, сколько воды поместится в "бассейн" с глубиной дна  $z$  и с высотой ограждающего бордюра  $p$ :

Объем =  $(p - z) \times$  количество просмотренных элементов в очереди.

Добавляем полученный объем к ранее найденному объему воды, заполняющему матрицу до высоты  $x$ .

Заполняем в матрице элементы, соответствующие просмотренным элементам из очереди, значением  $p$  ("Доливаем" воды в "бассейн" до уровня  $p$ ).

Переход на пункт а).

Суммарный полученный объем воды и является искомым.

Окаймление матрицы, упомянутое выше, может выглядеть следующим образом: матрица  $[1..N, 1..M]$  окаймляется нулями так: вводятся дополнительные нулевые 0-ая и  $(N+1)$ -ая строки и 0-ой и  $(M+1)$ -ый столбцы

```

var A: array[0..N+1,0..M+1] of byte;
{ ввод и окаймление нулями}
for i:=1 to N do begin
A[i,0]:=0; A[i,M+1]:=0;
for j:=1 to M do read(A[i,j]);
end;
for j:=0 to M+1 do begin
A[0,j]:=0; A[N+1,j]:=0;
end;

```

### Задача 15.

Легко заметить, что для существования решения необходимо проводить людей от самого левого входа к самому левому выходу, от самого правого – к правому и т.д. Поэтому приведем решение только для случая б).

Основная стратегия человека, вошедшего в самый левый вход, состоит в прохождении лабиринта, используя наиболее левые свободные места лабиринта, т.е. он должен двигаться, держась правой рукой за "стенку" лабиринта. Этот процесс можно формализовать следующим образом. Находясь в очередной позиции лабиринта, он должен помнить, с какой стороны он пришел сюда (слева, справа, сверху, снизу), и, руководствуясь этой информацией, выбрать следующее наиболее предпочтительное направление в новую позицию (куда он может пойти и где еще не был). При этом удобно использовать стек, в верхушке которого находятся координаты текущей позиции и направление, по которому в нее пришли. При этом все посещенные клетки метятся. Легко заметить, что если в позицию попали сверху, то наилучшим направлением будет налево, затем вниз, направо, и наконец назад (вверх). Аналогично можно определить наилучшие направления для других случаев.

Эта стратегия повторяется каждым из людей, при этом позиции, помеченные предыдущими людьми, считаются запрещенными для следующих.

### Задача 16.

Так как могут быть потребованы подписи лишь непосредственных подчиненных, то для любого разумного способа получения лицензии подпись всех подписавших чиновников, кроме первого, используются в одном и только одном наборе. Таким образом, мы можем определить стоимость подписи любого

чиновника как минимум по соответствующим ему набором сумм стоимостей подчиненных и взятии для набора. Таким образом, мы получаем простой и эффективный рекурсивный алгоритм определения стоимостей чиновников.

Но у этой задачи есть маленький подводный камень, а именно, должно быть заведено глобальное множество уже пройденных чиновников, что исключает повторную обработку этих чиновников. Сложность алгоритма - линейна по суммарному числу элементов в наборе.

## ЗАДАЧИ НА ГРАФАХ.

Предварительные сведения.

Что такое граф?

Часто в задачах встречается следующая конструкция - есть дома и дороги, их соединяющие; у каждой дороги есть длина. По другой терминологии такая конструкция называется графом, дома называются "вершинами", дороги - "ребрами" или "дугами", а длина дороги - "весом ребра" или "весом дуги". Таким образом фраза 'Найти минимальный вес пути между вершинами  $s$  и  $k$  в графе' может быть переведена так: 'Есть дома и дороги их соединяющие. Также заданы длины дорог. Найти кратчайшую длину пути от города  $s$  до города  $k$ , если двигаться можно только по дорогам'. Пропускная способность дуги  $(i,j)$  означает, например, сколько груза может быть перевезено по дороге (по дуге)  $(i,j)$  за единицу времени; а поток по дуге  $(i,j)$  - это сколько перевозится сейчас на самом деле).

Часто используют следующие обозначения:  $\Gamma(x(i))$ - множество вершин, в которые есть дуга из вершины  $i$ ;  $D(x(i))$ - множество вершин, из которых есть дуга в вершину  $i$ .

Пусть в графе  $N$  вершин.

Длины дуг обычно заносятся в матрицу (назовем ее  $C$ ) размера  $N$  на  $N$ , называемой матрицей смежности:

var C: array [1..N,1..N] of integer;

Элемент  $C[i,j]$  этой матрицы равен длине дуги (дороги), соединяющей вершины  $i$  и  $j$ , и равен (например) 0 или -1, если такой дуги нет. Если дорога двунаправленная (дуга неориентированная), то очевидно, что  $C[i,j]=C[j,i]$ .

### Алгоритм расстановки пометок для задачи о максимальном (от $s$ к $t$ ) потоке.

А. Расстановка пометок. Вершина может находиться в одном из трех состояний: вершине приписана пометка и вершина просмотрена (т.е. она имеет пометку и все смежные с ней вершины "обработаны"), пометка приписана, но вершина не просмотрена (т.е. она имеет пометку, но не все смежные с ней вершины обработаны), вершина не имеет пометки. Пометка произвольной вершины  $x(i)$  состоит из двух частей и имеет один из двух видов:  $(+x(j), m)$  или  $(-x(j), m)$ . Часть  $+x(j)$  пометки первого типа означает, что поток допускает увеличение вдоль дуги  $(x(j), x(i))$ . Часть  $-x(j)$  пометки второго типа означает, что поток может быть уменьшен вдоль дуги  $(x(i), x(j))$ . В обоих случаях  $m$  задает максимальную величину дополнительного потока, который может протекать от  $s$  к  $x(i)$  вдоль построенной увеличивающей цепи потока. Присвоение пометки вершине  $x(i)$  соответствует нахождению увеличивающей цепи потока от  $s$  к  $x(i)$ . Сначала все вершины не имеют пометок.

Шаг 1. Присвоить вершине  $s$  пометку  $(+s, m(s)=\text{бесконечность})$ . Вершине  $s$  присвоена пометка и она просмотрена, все остальные вершины без пометок.

Шаг 2. Взять некоторую непросмотренную вершину с пометкой; пусть ее пометка будет  $(+x(k), m(x(i)))$  ( $\pm$  обозначает, что перед  $x(k)$  может стоять как плюс, так и минус).

(I) Каждой помеченной вершине  $x(j)$ , принадлежащей  $\Gamma(x(i))$ , для которой  $c(i,j)<q(i,j)$ , присвоить пометку  $(-x(i), m(x(j)))$ , где

$$m(x(j))=\min[m(x(i)), q(i,j)-c(i,j)].$$

(II) Каждой непомеченной вершине  $x(j)$ , принадлежащей  $D(x)$ , для которой  $c(i,j)>0$ , присвоить пометку  $(-x(i), m(x(j)))$ , где

$m(x(j))=\min[m(x(i)), c(j,i)]$ . (Теперь вершина  $x(i)$  и помечена, и просмотрена, а вершины  $x(j)$ , пометки которым присвоены в (I) и (II), являются непросмотренными.) Обозначить каким-либо способом, что вершина  $x(i)$  просмотрена.

Шаг 3. Повторять шаг 2 до тех пор, пока либо вершина  $t$  будет помечена, и тогда перейти к шагу 4, либо  $t$  будет не помечена и никаких других пометок нельзя будет расставить; в этом случае алгоритм заканчивает работу с максимальным вектором потока  $s$ . Здесь следует отметить, что если  $X(0)$  – множество помеченных вершин, а  $X'(0)$  – множество не помеченных, то  $(X(0) \rightarrow X'(0))$  является минимальным разрезом.

Б. Увеличение потока.

Шаг 4. Положить  $x=t$  и перейти к шагу 5.



Шаг 5. (I) Если пометка в вершине  $x$  имеет вид  $(+z, m(x))$ , то изменить поток вдоль дуги  $(z, x)$  с  $c(z, x)$  на  $c(z, x) + m(t)$ .

(II) Если пометка в вершине  $x$  имеет вид  $(-x, z)$  с  $c(x, z)$  на  $c(x, z) - m(t)$ .

Шаг 6. Если  $z=s$ , то стереть все пометки и вернуться к шагу 1, чтобы начать расставлять пометки, но используя уже улучшенный поток, найденный на шаге 5. Если  $z < s$ , то взять  $x=z$  и вернуть к шагу 5.

Обозначения:  $\Gamma(x(i))$  – множество вершин, в которые есть дуга из вершины  $i$ ;

$D(x(i))$  – множество вершин, из которых есть дуга в вершину  $i$ ;

$c(i, j)$  – это пропускная способность дуги (т.е., например, сколько груза может быть перевезено по дороге (по дуге)  $(i, j)$  за единицу времени);

$q(i, j)$  – поток по дуге  $(i, j)$  (т.е. сколько перевозится сейчас на самом деле).

### **Кратчайшее расстояние от вершины начало до остальных вершин.**

(Алгоритм Дейкстры). Обозначения:

$C[i, j]$  – длина ребра  $(i, j)$ ,  $C[i, j] \geq 0$  (если ребра нет, то его длина полагается равной бесконечности).

$D[i]$  – кратчайшее текущее расстояние от вершины нач до вершины  $i$ .

флаг[ $i$ ] – информация о просмотре вершины  $i$ : 0 – если вершина не просмотрена, 1 – если просмотрена.

Если вершина просмотрена, то для нее  $D[i]$  есть наикратчайшее расстояние от вершины нач до вершины  $i$ .

предок[ $i$ ] – информация о номере вершины, предшествующей вершине  $i$  в кратчайшем пути от вершины нач.

минрас – это минимальное расстояние.

Алгоритм:

для  $i$  от 1 до  $N$  выполнять

нц

предок[ $i$ ]:=нач;

флаг[ $i$ ]:=0;

$D[i]:=C[\text{нач}, i]$

кц

флаг[нач]:=1; { пока мы знаем только расстояние }

предок[нач]:=0 { от вершины нач до нее же, равное 0 }

для  $i$  от 1 до  $N-1$  выполнять

нц

минрас:=бесконечность;

для  $j$  от 1 до  $N$  выполнять

если (флаг[ $j$ ]=0 и (минрас  $>$   $D[j]$ ) { находим минимальное }

то минрас:= $D[j]$ ; {расстояние }

$k:=j$ ; { до непомеченных вершин }

все

флаг[ $k$ ]:=1; { вершина  $k$  помечается просмотренной }

для  $j$  от 1 до  $N$  выполнять { выполняем просмотр }

если флаг[ $j$ ]=0 и  $D[j] > D[k] + C[k, j]$

{ Т.е. если для вершины  $j$  еще не найдено кратчайшее

расстояние от нач, и из вершины  $k$  по дуге  $C[k, j]$

путь в  $j$  короче, чем найденный ранее }

то  $D[j]:=D[k] + C[k, j]$  { то запоминаем его }

предок[ $j$ ]:= $k$ ;

все

кц

## УСЛОВИЯ ЗАДАЧ.

### Задача 0.

Задан набор неповторяющихся пар  $(A_i, A_j)$ ,  $A_i, A_j$  принадлежат множеству  $A = \{A_1, A_2, \dots, A_n\}$ . Необходимо составить цепочку максимальной длины по правилу

$$(A_i, A_j) + (A_j, A_k) = (A_i, A_j, A_k).$$

При образовании этой цепочки любая пара может быть использована не более одного раза.

### Задача 1.

Между  $N$  пунктами ( $N \leq 50$ ) заданы дороги длиной  $A(i, j)$ , где  $i, j$  – номера пунктов. Дороги проложены на разной высоте и пересекаются только в общих пунктах. В начальный момент времени из заданных пунктов начинают двигаться с постоянной скоростью  $M$  роботов ( $M=2$  или  $3$ ), независимо меняя направление движения только в пунктах. Роботы управляются таким образом, чтобы минимизировать время до встречи всех роботов в одном месте. Скорость  $i$ -того робота может быть равна 1 или 2. Остановка роботов запрещена.

Задание:

Написать программу, которая:

1) при заданных  $N, M$  и сети дорог единичной длины (все имеющиеся  $A(i, j)=1$ ) определяет минимальное время, через которое может произойти встреча всех  $M$  роботов, при этом начальное положение роботов и скорость их движения известны.

2) Выполнить те же действия, что и в п. 1, но только для различных значений  $A(i, j)$ . Примечание: В случае невозможности встречи всех  $M$  роботов в одном месте ни в какой момент времени в результате выполнения программы должно быть сформировано соответствующее сообщение.

Требование к вводу-выводу:

- 1) все входные данные – целые неотрицательные числа;
- 2) при задании сети дорог должно быть указано количество дорог –  $K$  и пункты их начала и конца в виде пар  $(i, j)$ .

### Задача 2.

На плоскости расположено  $N$  точек. Имеется робот, который движется следующим образом. Стартуя с некоторой начальной точки и имея некоторое начальное направление, робот движется до первой встреченной на его пути точки, изменяя в ней свое текущее направление на 90 градусов, т.е. поворачивая налево или направо. После этого он продолжает движение аналогично. Если робот достиг начальной точки, либо не может достичь новой точки (которую он еще не посещал), то он останавливается.

Определить, может ли робот посетить все  $N$  точек, если:

1. Определены начальная точка и направление робота.
2. Определена начальная точка, а направление робота можно выбирать.
3. Начальную точку и направление робота можно выбирать.

Координаты точек – целые числа, угол измеряется в радианах относительно оси  $Ox$ .

### Задача 3. "ПУТЬ".

Найти кратчайшее расстояние между двумя вершинами в графе. Найти все возможные пути между этими двумя вершинами в графе непересекающиеся по

- а) ребрам
- б) вершинам.

### Задача 4.

Лабиринт задается матрицей смежности  $N \times N$ , где  $C(i, j)=1$ , если узел  $i$  связан узлом  $j$  посредством дороги. Часть узлов назначается входами, часть – выходами. Входы и выходы задаются последовательностями узлов  $X(1), \dots, X(p)$  и  $Y(1), \dots, Y(k)$  соответственно.

Найти максимальное число людей, которых можно провести от входов до выходов таким образом,

чтобы:

- а) их пути не пересекались по дорогам, но могут пересекаться по узлам;
- б) их пути не пересекались по узлам;

### Задача 5.

$N$  шестеренок пронумерованы от 1 до  $N$  ( $N \leq 10$ ). Заданы  $M$  ( $0 \leq M \leq 45$ ) соединений пар шестеренок в виде  $(i, j)$ ,  $1 \leq i < j \leq N$  (шестерня с номером  $i$  находится в зацеплении с шестерней  $j$ ). Можно ли повернуть шестерню с номером 1?

Если да, то найти количество шестерен, пришедших в движение.

Если нет, то требуется убрать минимальное число шестерен так, чтобы в оставшейся системе при вращении шестерни 1 во вращение пришло бы максимальное число шестерен. Указать номера убранных шестерен (если такой набор не один, то любой из них) и количество шестерен, пришедших в движение.

### Задача 6.

Имеется  $N$  прямоугольных конвертов и  $N$  прямоугольных открыток различных размеров. Можно ли разложить все открытки по конвертам, чтобы в каждом конверте было по одной открытке. Замечание. Открытки нельзя складывать, сгибать и т.п., но можно помещать в конверт под углом. Например, открытка с размерами сторон 5:1 помещается в конверты с размерами 5:1, 6:3, 4.3:4.3, но не входит в конверты с размерами 4:1, 10:0.5, 4.2:4.2.

### Задача 7.

Составить программу для нахождения произвольного разбиения 20 студентов на 2 команды, численность которых отличается не более чем в 2 раза, если известно, что в любой команде должны быть студенты, обязательно знакомые друг с другом. Круг знакомств задается матрицей  $(20, 20)$  с элементами

$$A(ij) = \begin{cases} 1, & \text{если } i \text{ студент знаком с } j \\ 0, & \text{иначе} \end{cases}$$

### Задача 8.

Имеется  $N$  человек и прямоугольная таблица  $A[1:N, 1:N]$ ; элемент  $A[i, j]$  равен 1, если человек  $i$  знаком с человеком  $j$ ,  $A[i, j] = A[j, i]$ . Можно ли разбить людей на 2 группы, чтобы в каждой группе были только незнакомые люди.

### Задача 9.

На олимпиаду прибыло  $N$  человек. Некоторые из них знакомы между собой. Можно ли опосредованно познакомиться их всех между собой? (Незнакомые люди могут познакомиться только через общего знакомого).

### Задача 10.

Пусть группа состоит из  $N$  человек. В ней каждый имеет  $(N/2)$  друзей и не больше  $K$  врагов. У одного из них есть книга, которую все хотели бы прочитать и потом обсудить с некоторыми из остальных.

Написать программу, которая:

1. Находит способ передачи книги таким образом, чтобы она побывала у каждого в точности один раз, переходя только от друга к другу и наконец возвратилась к своему владельцу.
2. Разбивает людей на  $S$  групп, где будет обсуждаться книга, таким образом, чтобы вместе с каждым человеком в ту же самую группу вошло не более  $P$  его врагов.

Примечание: предполагается, что  $S \cdot P \geq K$ .

### Задача 11.

В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа ровно один раз.

### Задача 12.

$N$  различных станков один за другим объединены в конвейер. Имеется  $N$  рабочих. Задана матрица  $C[N,N]$ , где  $C[i,j]$  производительность  $i$ -ого рабочего на  $j$ -ом станке. Определить

- на каком станке должен работать каждый из рабочих, чтобы производительность была максимальной.
- то же, но станки расположены параллельно и выполняют однородные операции.

### Задача 13.

Имеется  $N$  городов. Для каждой пары городов  $(I,J)$  можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги  $A(I,J)$ . Вне городов дороги не пересекаются.

Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей  $B[1:N,1:N]$ , где  $B[I,J]=1$  тогда и только тогда, когда дорогу, соединяющую города  $I$  и  $J$ , следует строить.

### Задача 14.

Вводится  $N$  – количество домов и  $K$  – количество дорог. Дома пронумерованы от 1 до  $N$ . Каждая дорога определяется тройкой чисел – двумя номерами домов – концов дороги и длиной дороги. В каждом доме живет по одному человеку. Найти точку – место встречи всех людей, от которой суммарное расстояние до всех домов будет минимальным.

Если точка лежит на дороге, то указать номера домов – концов этой дороги и расстояние от первого из этих домов. Если точка совпадает с домом, то указать номер этого дома.

Примечание: длины дорог – положительные целые числа.

### Задача 15.

$N$  колец сцеплены между собой (задана матрица  $A(n*n)$ ,  $A(i,j)=1$  в случае, если кольца  $i$  и  $j$  сцеплены друг с другом и  $A(i,j)=0$  иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.

### Задача 16.

Янка положил на стол  $N$  выпуклых  $K$ -гранников и  $N$  различных типов наклеек по  $K$  штук каждая. Ночью кто-то наклеил наклейки на грани, по одной на грань.

Помогите Янке расставить многогранники так, чтобы наклейка каждого типа была видна ровно  $K-1$  раз.

### Задача 17.

Имеется  $N$  точек и  $M$  проводков. Проводком можно соединить некоторую пару различных точек, причем пара может быть соединена не более чем одним проводком. Все проводки должны быть использованы.

Пусть  $D_i$  – количество проводков, которые будут соединены с точкой с номером  $i$ ,  $i=1, \dots, N$ .

Необходимо соединить  $N$  точек с помощью  $M$  проводков таким образом, чтобы сумма  $S=D_1*D_1 + D_2*D_2 + \dots + D_n*D_n$  была максимальной.

Вывести величины  $D_i$  в неубывающем порядке и. по требованию (priznak=1), список соединений.

ВВОД:

<Введите N:> N ( $N \leq 100$ )

<Введите M:> M ( $M \leq 1000$ )

<PRIZNAK=> PRIZNAK ВЫВОД:

<Результирующая конфигурация:>  $D_i$  в неубывающем порядке.

<Сумма S> S

<Список соединений>

<Точку 1 соединить с> список точек

.....

<Точку N соединить с> список точек

### Задача 18.

Задано  $N$  городов с номерами от 1 до  $N$  и сеть из  $M$  дорог с односторонним движением между ними. Каждая дорога задается тройкой  $(i, j, k)$ , где  $i$  – номер города, в котором дорога начинается,  $j$  – номер города, в котором дорога заканчивается, а  $k$  – ее длина (число  $k$  – натуральное). Дороги друг с другом могут пересекаться только в концевых городах.

Все пути между двумя указанными городами  $A$  и  $B$  можно упорядочить в список по неубыванию их длин (если есть несколько путей одинаковой длины, то выбираем один из них). Необходимо найти один из путей, который может быть вторым в списке.

Вывести его длину  $L$  и города, через которые он проходит.

ВВОД:

<Количество городов  $N$ :>  $N$

<Количество дорог  $M$ :>  $M$

<Начало, конец и длина дороги 1:>  $i_1, j_1, k_1$

.....

<Начало, конец и длина дороги  $M$ :>  $i_M, j_M, k_M$

<Города  $A$  и  $B$ , между которыми надо найти путь:>  $A, B$  ВЫВОД:

<Пути нет>

или

<Путь проходит по городам>  $A, i_1, i_2, \dots, B$

<Длина пути>  $L$

## РЕШЕНИЕ ЗАДАЧ НА ГРАФАХ.

### Задача 0.

Для более удобного хранения информации заведем матрицу  $C[1..n, 1..n]$  (так называемую матрицу смежности) в которой  $C[i, j]=1$ , если в наборе есть пара  $(A_i, A_j)$  и  $C[i, j]=0$  иначе.

Будем строить все возможные цепочки (по правилу, данному в условии) и искать среди них ту, которая имеет максимальную длину.

В качестве начального символа цепочки можно взять любой символ из  $A$ . Пусть это символ  $A_i$ . Ищем, просматривая строку  $i$  матрицы  $C$  слева направо элемент  $C[i, j]=1$  (другими словами, ищем пару с первым элементом  $A_i$ ). Если такого элемента не существует, то берем в качестве начала строки другой элемент множества  $A$ . Если элемент  $C[i, j]=1$  найден, то ему соответствует пара  $(A_i, A_j)$ . Помечаем ее как уже использованную полагая, например,  $C[i, j] = -1$ . Далее просматриваем слева направо строку  $j$  матрицы  $C$  в поисках еще не использованной пары  $(A_j, A_k)$  ( $C[j, k]=1$ ). Присоединяем элемент  $A_k$  к имеющейся цепочке, полагая  $C[j, k] = -1$ , ищем единичный элемент в строке  $k$  и т.д. Предположим, на некотором шаге мы получили цепочку  $A_i A_j A_k \dots A_s A_l A_p$  и в строке  $p$  матрицы больше нет ни одного единичного элемента. Это означает, что при таком подборе предыдущих элементов мы нашли максимальную по длине строку. Если ее длина больше длин всех найденных ранее строк, запоминаем эту строку как рекорд. После этого "отщепляем" от строки последний элемент  $A_p$  и смотрим, есть ли еще в строке  $l$  единичный элемент с индексом, большим  $p$ . Если да, то приписываем уже этот элемент к строке и пытаемся затем снова увеличить длину полученной строки, если же нет, то "отщепляем" от строки элемент  $A_l$ , в строке  $S$  ищем единичный элемент с индексом, большим  $l$  и т.д.

Останов осуществляется тогда, когда мы должны "отщепить" от строки  $A_i$ .

Перебираем цепочки, начинающиеся со всех возможных элементов множества  $A$ . Находим строку максимальной длины:

```
const M=10; { максимально число элементов в A }
{ будем считать, что A состоит из чисел от 1 до N }
var c:array[1..M,1..M] of integer;
curstr, maxstr: array[0..M] of integer;
{ в этих переменных хранятся текущая цепочка и }
{ цепочка максимальной длины. }
{ В нулевом элементе хранится длина цепочки }
N,E : integer; { N - число элементов в A }
i,j,k : integer; { E - число пар в наборе }
procedure find;
var l,j : integer;
begin
l:=curstr[curstr[0]]; { l = последний элемент цепочки }
for j:=1 to N do { просмотр строки l }
if C[l,j]=1
then begin
curstr[0]:=curstr[0]+1;
curstr[curstr[0]]:=j; { j -> в цепочку }
c[l,j]:=-1; { пара использована }
find;
c[l,j]:=1; { пару снова разрешено }
{ использовать }
curstr[0]:=curstr[0]-1;
end;
if curstr[0]>maxstr[0] { если нашли более }
then maxstr:=curstr { длинную строку }
end;
begin
```

```

readln(N); readln(E);
for i:=1 to N do
  for j:=1 to N do
    C[i,j]:=0;
  for k:=1 to E do begin
    write('очередная пара: ',i,j);
    c[i,j]:=1
  end;
  for i:=1 to N do begin
    curr[0]:=1; { поиск цепочки }
    curr[1]:=i; { начинающейся элементом i }
    find;
  end;
  for i:=1 to maxstr[0] do
    write(maxstr[i]); { печать максимальной строки }
  end.

```

### Задача 1.

Для решения задачи важно понять, что встреча роботов может произойти либо в пункте, либо на дороге. В первом случае задача решается просто: необходимо последовательно строить множества пунктов для каждого робота, в которых они могут оказаться через время, равное 0.5, 1, 1.5, 2, ..., T. Это можно делать, используя очереди. В случае же встречи роботов на дороге легко понять, что непосредственно перед встречей все они должны были находиться в следующих пунктах:

1. Либо в двух пунктах, связанных дорогой;
2. Либо в пунктах, из которых есть дороги в один и тот же пункт;
3. Либо в трех пунктах, образующих треугольник.

Поэтому, после каждого целого такта времени, достаточно проверять, находятся ли роботы в одной из описанных 3 ситуаций. При этом время подсчитывается очевидным способом.

### Задача 2.

Рассмотрим только случай, когда роботы двигаются только параллельно координатным осям, в других случаях можно сделать преобразование координат.

Легко понять, что если рассмотреть точку, имеющую наибольшую координату Y, причем самую левую (т.е. наименьшую координату X, если таких несколько), то для нее существует только 2 возможности быть пройденной роботом: робот должен прийти из ближайшей точки A снизу и пойти в ближайшую точку B справа или наоборот. В любом случае эти 2 отрезка фиксированы для робота. Теперь те же рассуждения можно провести и для точек B и точки, находящейся правее B, а также для самых нижних, левых и правых точек. Окончательно получается, что возможный проход робота строго фиксирован. Если упорядочить точки по горизонталям (вертикалям), то первая точка горизонтали (вертикали) должна соединяться со второй, третья с четвертой и т.д. Понятно, что если получившаяся фигура связна (цикл), то решение существует для случаев 2. и 3., а для случая 1. нужно проверить, в нужном ли направлении стоит робот. Однако есть одна трудность в случае, когда существуют горизонталь и вертикаль, содержащие нечетное число точек, а обход существует. Это возможно только в случае, когда это стартовая точка обхода, причем она находится в 'нечетной' горизонтали и вертикали. Удалив ее, можно воспользоваться предыдущей процедурой, при этом фиксированные отрезки не должны пересекаться в удаленной точке.

### Задача 3.

Для решения задачи достаточно воспользоваться алгоритмом нахождения потока между двумя заданными вершинами, преобразуя в случае а) граф по следующему правилу: каждую вершину исходного графа превращаем в ребро с пропускной способностью 1.

#### Задача 4.

Решается аналогично задаче 3.

#### Задача 5.

Будем обозначать вращение по часовой стрелке нулем, против - единицей. Сначала припишем шестерне с номером 1 число нуль. На следующем шаге всем шестерням, сцепленным с первой, будут приписаны числа 1 (они будут вращаться в противоположную шестерне 1 сторону). Далее всем шестерням, находящимся в зацеплении с занумерованными на предыдущем шаге, припишем значения 0 и т.д.

Процесс будем повторять до тех пор, пока либо на очередном шаге ни одной шестерне не будет приписано новое значение, и тогда шестерню с номером 1 удастся повернуть, и число рассмотренных шестеренок и есть искомое число пришедших в движение, либо на каком-то шаге пометка шестерни изменяется с 1 на 0 или с 0 на 1, и тогда система в движение не придет.

Перебирая сначала все возможные варианты выбрасывания по одной, затем, в случае неудачи по две, три, ... ,  $N-1$  шестерни и проводя аналогичные рассуждения мы получаем максимальное число шестерен, которое могло бы прийти в движение. О генерации вариантов выбрасывания шестерен см. задачу 1 (перебор).

#### Задача 6.

Можно сформировать граф, состоящий из  $2*N$  вершин, соответствующих открыткам и конвертам, причем две вершины соединены ребром, если одна соответствует открытке, а другая – конверту, при этом соответствующая открытка входит в соответствующий конверт. Добавив в этом графе 2 новые вершины, одна из которых смежная всем вершинам, соответствующим открыткам, а другая – всем вершинам, соответствующим конвертам, сведем задачу к задаче нахождения максимального потока между этими вершинами.

#### Задача 7.

Разбиение на группы осуществляется аналогично, как и в Задаче 8. за тем исключением, что в ситуации 3. организуются две новые группы (пара групп, одна из них может оказаться в результате пустой). После разбиения всех людей будем использовать принцип формирования двух результирующих групп, основываясь на идее решения Задачи о распределении камней на две кучки (очередной камень берется в группу с меньшим весом)), учитывая возможности объединения двух пар групп в одну пару групп произвольным слиянием двух групп из различных пар.

#### Задача 8.

Задача решается следующим образом. Выбирается произвольный человек и помещается в первую группу. Затем находятся люди, ему знакомые. Понятно, что они не могут находиться в первой группе, поэтому их необходимо поместить во вторую. Затем находятся люди, знакомые людям из второй группы. Понятно, что они не могут находиться во второй группе, поэтому их необходимо поместить в первую. Повторяя так поочередно для каждой из групп, мы придем к одной из ситуаций:

1. Какой-то человек сначала был помещен в одну группу, а потом должен быть помещен в другую. Понятно, что в этом случае

задача не имеет решения.

2. Каждый человек помещен в одну из групп. В этом случае задача решена.

3. Не каждый человек помещен в одну из групп. Это означает, что оставшиеся не помещенные в группы люди не знакомы людям в группах (иначе они были бы куда-нибудь помещены). Следовательно, одного из них безразлично куда помещать, например в первую группу. Затем описанный выше процесс продолжается, пока не придем к ситуации 1. или 2.

#### Задача 9.

Для решения задачи достаточно определить, является ли связным граф, определяемый матрицей



смежности, элементы которой  $a[i,j]$  равны 1, если люди с номерами  $i$  и  $j$  знакомы и равны 0 иначе. Граф называется связным, если существует путь между любыми парами его вершин. Понятно, что путь между вершинами  $i$  и  $j$  в таком графе и определяет возможную последовательность знакомств, позволяющих познакомить людей с номерами  $i$  и  $j$ .

Для определения связности графа можно воспользоваться методом поиска в ширину, который состоит в следующем.

На начальном этапе в очередь помещается некоторая начальная вершина, например вершина с номером 1.

На каждой из следующих итераций (пока очередь не пуста) выполняются следующие действия:

- извлекается вершина из очереди;
- определяются вершины, ей смежные и которые в очереди еще не были, и помещаются в очередь.

Если в результате таких действий все вершины побывали в очереди (а для этого удобнее подсчитывать количество вершин, там побывавших), то граф связан, иначе не связан. Для маркировки вершин, побывавших в очереди, можно использовать массив размера  $N$  с элементами 0 и 1.

### Задача 10.

Задача может быть сформулирована в графовой постановке следующим образом: найти простой цикл в графе (т.е. без повторяющихся вершин), проходящий через все вершины графа. В общем случае не существует эффективного алгоритма решения этой задачи. Однако в данном случае задачу можно решить эффективно.

Предположим, что уже построен некоторый простой путь  $(x[1], x[2], \dots, x[k])$  Множество вершин, вошедших в путь, будем считать пройденными, остальные не пройденные. Возможны 3 ситуации.

1. Одна из вершин  $x[1], x[k]$  смежна одной из не пройденных еще вершин. В этом случае путь можно очевидным образом удлинить на одну вершину.

2. Ни одна из вершин  $x[1], x[k]$  не смежна одной из не пройденных еще вершин, а вершины  $x[1]$  и  $x[k]$  смежны. В этом случае понятно, что  $k > N/2$ , так как вершины  $x[1]$  и  $x[k]$  смежны  $N/2$  вершинам. Следовательно количество не пройденных вершин не больше  $N/2$ . Следовательно любая вершина  $u$  из них смежна одной из пройденных вершин, например  $x[i]$ . Но тогда можно получить более длинный путь  $(u, x[i], x[i+1], \dots, x[k], x[1], x[2], \dots, x[i-1])$ .

3. В этом случае степеней вершин нетрудно показать, что в пути  $(x[1], x[2], \dots, x[k])$  существует такой индекс  $i$ , что  $x[1]$  смежна  $x[i]$ , а  $x[i-1]$  смежна  $x[k]$ . Следовательно, рассмотрев путь  $(x[i], x[i+1], \dots, x[k], x[i-1], x[i-2], \dots, x[1])$  мы имеем ситуацию 2, поэтому можно получить более длинный путь.

Применяя описанный выше способ начиная с пути длины 1, построим простой цикл, включающий все вершины.

### Задача 11.

В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа ровно один раз.

Для существования такого цикла необходимо и достаточно наличия двух условий:

1. граф является связным;
2. степень любой вершины графа четна. Первое условие очевидно. Второе следует из факта, что если такой цикл существует и заходит в некоторую вершину графа, то он должен и выйти из нее. Проверив эти условия (для проверки связности можно воспользоваться алгоритмом для задачи 10).

Предположим, что некоторый цикл  $A$  (не обязательно проходящий через все ребра графа) уже построен и из графа выброшены ребра цикла (их можно просто отметить как пройденные). Найдем вершину (пусть это вершина  $k$ ), через которую этот цикл проходил, но которой инцидентны некоторые ребра (непройденные). Построим новый цикл  $B$ , который начинается в вершине  $k$ , по следующему правилу.

Находясь в текущей вершине цикла ( вначале это вершина  $k$ ), находим непройденное еще ребро, инцидентное текущей вершине. Это ребро и определяет новую текущую вершину цикла, а ребро считается пройденным. Построение цикла  $B$  продолжается до тех пор, пока это возможно. Легко понять, что это будет цикл, так как степень любой вершины четна. Предположим, что построение цикла  $B$  завершено. Склеиваем теперь циклы  $A$  и  $B$  следующим образом. Находим в цикле  $A$  (последовательности вершин)

вершину  $k$  и заменяем ее последовательностью вершин цикла  $B$ .

Такой процесс повторяется до тех пор, пока все ребра не будут пройдены. Описанный процесс можно начинать с пустого цикла  $A$  (пустой последовательности).

### Задача 12.

Производительность конвейера определяется минимальной производительностью рабочего на конвейере. Поэтому для решения задачи нам достаточно определить максимальную производительность  $P$ , при которой возможно распределить по станкам, всех рабочих, когда для каждого рабочего определен набор станков, на которые он может быть поставлен (понятно, что это те станки, производительность на которых у рабочего не ниже производительности  $P$ ). Легко понять, что для выбранной производительности  $X$  задача определения, возможно ли распределение по станкам для каждого рабочего, аналогична решению задачи 4. Для определения максимальной производительности  $P$ , при которой возможно распределить по станкам всех рабочих, можно воспользоваться методом дихотомии следующим образом. Отсортируем в одномерном массиве размера  $N \times N$  все элементы  $C[i, j]$ ,  $1 \leq i, j \leq N$ . Определим левую границу  $ЛГ=1$  и правую границу  $ПГ=N$ .

Для элемента массива  $A$  с индексом  $СГ=(ЛГ+ПГ)/2$ , рассматриваемого в качестве возможной максимальной производительности  $P$ , определяем, можно ли распределить по станкам всех рабочих. Если да, то величина возможной максимальной производительности  $P$  увеличивается. Для этого пересчитывается значение левой границы по правилу  $ЛГ:=СГ+1$ . Если нет, то величина возможной максимальной производительности  $P$  уменьшается. Для этого пересчитывается значение правой границы по правилу  $ПГ:=СГ-1$ .

Процесс заканчивается, когда  $ЛГ \geq ПГ$ . Значение  $A[ПГ-1]$  и является максимальной производительностью  $P$ , а конкретное распределение рабочих определяется структурой максимального потока.

### Задача 13.

Легко понять, что сеть дорог будет реализовывать некоторый связный (так как можно проехать из любого города в любой) граф без циклов (так как одно ребро из цикла можно выбросить, а связный граф останется связным). Поэтому алгоритм построения сети дорог минимальной суммарной стоимости очень прост. На каждой итерации необходимо находить дорогу минимальной стоимости, которая не образует цикла с уже выбранными дорогами на предыдущих итерациях. Основную трудность такого решения составляет проверка условия, образуют ли ребра цикл.

Однако решение существенно упрощается, если рассматривать только минимальные ребра только между двумя множествами: множеством помеченных вершин и множеством непомеченных вершин. Понятно, что эти множества должно соединять хотя бы одно ребро, чтобы граф был связным. Ясно, что оно должно быть минимальным по длине. В описываемом ниже алгоритме это делается следующим образом.

Для каждой вершины  $k$  из множества непомеченных вершин (а на начальном этапе это все вершины, кроме первой) определяется ближайшая вершина из множества помеченных вершин  $БЛИЖ[k]$ . На каждой итерации определяется кратчайшее ребро  $(i, j)$  между множеством помеченных вершин и множеством непомеченных вершин, используя массив  $БЛИЖ$ . Найденное ребро выбирается для системы дорог, а соответствующая вершина  $j$  считается помеченной. После этого пересчитывается массив  $БЛИЖ$ . При этом учитывается, что  $k$  изменение некоторой величины  $БЛИЖ[k]$  может произойти только тогда, когда расстояние от  $k$  до  $j$  меньше, чем от  $k$  до  $БЛИЖ[k]$ .

Алгоритм

для  $i$  от 1 до  $N$  выполнять

нц

флаг[ $i$ ]:=0;

$БЛИЖ[i]:=1$

кц

флаг[1]:=1;

для  $k$  от 1 до  $N-1$  выполнять

```

нц
минрас:=бесконечность;
для i от 2 до N выполнять
если флаг[i]=0 и минрас > C[БЛИЖ[i],i]
то минрас:=C[БЛИЖ[i],i];
j:=i;
все
Вывод ребра (БЛИЖ[j],j)
флаг[j]:=1;
для i от 2 до N выполнять
если флаг[i]=0 и C[БЛИЖ[i],i]>C[i,j]
то БЛИЖ[i]:=j;
все
кц

```

#### Задача 14.

Предположим, что мы нашли точку встречи  $z$ , и пусть она лежит на дороге  $(u,v)$  длины  $l$  на расстоянии  $d > 0$  от дома  $u$  и на расстоянии  $(l-d) > 0$  от дома  $v$ . Все множество домов разделим на два непересекающихся подмножества  $V$  и  $U$ . В подмножество  $U$  входят те дома, расстояние от которых до дома  $v$  меньше, чем расстояние до дома  $u$ . Все остальные дома отнесем к подмножеству  $U$ .

Пусть  $V$  подмножестве  $V$  домов  $K_v$ , а в  $U$  –  $K_u$ , и пусть  $K_v > K_u$ .

Обозначим суммарное расстояние от точки  $z$  (находящейся на расстоянии  $d$  от дома  $u$ ) до всех  $N$  домов через  $R(z, d)$ . Очевидно, что

$R(z, d) = \text{сумма расстояний от } v \text{ до домов множества } V + \text{сумма расстояний от } u \text{ до домов множества } U + K_u * d + K_v * (l - d)$ .

Если мы сдвинем  $z$  на расстояние  $p$ ,  $p < (l - d)$  по направлению к дому  $v$ , то  $R(z, d+p) = R(z, d) + K_u * p + K_v * (-p) = R(z, d) + (K_u - K_v)p < R(z, d)$  ?!

т.е. первоначальная установка точки  $z$  была неверной. Случай  $K_v < K_u$  рассматривается аналогично. Если же  $K_v = K_u$ , то точка  $z$  может быть на дороге в любом месте, в том числе и в концевых домах.

Итак, из всего выше сказанного следует, что искомая точка совпадает с одним из  $N$  домов, и что нам достаточно для каждого из домов  $i$  вычислить (например, по алгоритму Дейкстры) кратчайшее расстояние от него до каждого из оставшихся домов, затем найти сумму этих кратчайших расстояний (т.е. минимальное суммарное расстояние до всех домов от  $i$ ). Минимальное из суммарных расстояний по всем  $i$  и даст решение задачи.

#### Задача 15.

Предполагаемый вариант решения использует полный перебор всех вариантов. В качестве первого звена цепочки берем последовательно каждое из  $N$  колец. Ищем сцепленные с этим кольцом, выбираем среди них одно (все остальные кольца будут считаться удаленными – т.е. строки и столбцы с номерами этих колец должны обнуляться). С этим изменением задача сводится к задаче 0 (Графы). Действительно, если мы найдем максимальный путь в графе (в данном случае – максимальную длину цепочки в начальной конструкции), то мы минимизируем количество выброшенных колец.

#### Задача 16.

Можно сформировать граф, состоящий из  $2 * N$  вершин, соответствующих  $K$ -гранникам и типу наклеек, причем две вершины соединены ребром, если одна соответствует наклейке, а другая –  $K$ -граннику, при этом наклейка соответствующего типа находится на соответствующем  $K$ -граннике. Добавив в этот граф 2 новые вершины, одна из которых смежна всем вершинам, соответствующим типам наклеек, а другая – всем вершинам, соответствующим  $K$ -гранникам, сведем задачу к задаче нахождения максимального потока между этими вершинами.

### Задача 17.

Показывается, что точное решение есть максимум из двух решений, полученных при помощи следующих эвристик:

1) Вершину 1 соединяем последовательно с вершинами 2, 3, ..., N.

Вершину 2 соединяем последовательно с вершинами 3, 4, ..., N.

.....

Вершину  $i$  соединяем последовательно с вершинами  $i+1, i+2, \dots, N$ .

.....

и так до тех пор, пока хватит проводков.

2) Вершину 2 соединяем с вершиной 1.

Вершину 3 соединяем последовательно с вершинами 1, 2.

.....

Вершину  $i$  соединяем последовательно с вершинами  $i-1, i-2, \dots, 1$ .

.....

и так до тех пор, пока хватит проводков.

### Задача 18.

Алгоритм нахождения второго минимального расстояния между двумя вершинами:

По алгоритму Дейкстры находим кратчайший путь между начальной вершиной N и конечной K, состоящий из дуг  $a_1, \dots, a_s$ . Во втором по длине кратчайшем пути из N в K не будет по крайней мере одного из ребер  $a_i$ . Поэтому алгоритм нахождения этого второго пути будет следующим:

Для  $i$  от 1 до  $s$  повторять

удалить из графа ребро (дорогу)  $a_i$ ;

найти кратчайший путь из N в K в новом графе;

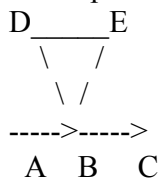
если его длина меньше рекордной минимальной длины,  
полученной ранее,

то запомнить текущий путь и его длину как рекорд;

восстановить в графе ребро (дорогу)  $a_i$ ;

конец\_для;

Этот алгоритм можно найти в книге Кристофидеса "Теория графов. Алгоритмический подход". Но в нем не рассматривается следующий случай:



Кратчайший путь – ABC, второй – ABDEBC.

Для того, чтобы его обработать, достаточно найти кратчайший нетривиальный путь из каждой вершины в нее же. Для этого в алгоритме Дейкстры достаточно на нулевом шаге не выставлять пометку "Просмотрено" на начальной вершине.

## Экзаменационные задачи.

---

1. За один проход (просмотр файла) вычислить сумму и произведение элементов, находящихся в файле "temp.dat".
2. За один проход (просмотр файла) вычислить номер первого и последнего максимального элемента, находящихся в файле "temp.dat".
3. За один проход (просмотр файла) вычислить число минимальных элементов, находящихся в файле "temp.dat".
4. За один проход (просмотр файла) вычислить число элементов, больших предыдущего, находящихся в файле "temp.dat".
5. Вычислить средне квадратичное отклонение от среднего арифметического элементов, находящихся в файле "temp.dat":  
$$s = \sqrt{((x_1 - M)^2 + \dots + (x_n - M)^2) / n}$$
  
(n - число элементов, M - среднее арифметическое).
6. За один проход (просмотр файла) вычислить номер первого и последнего элемента, равного числу x (если таких нет, считать номер равным нулю), находящихся в файле "temp.dat".
7. За один проход (просмотр файла) вычислить число различных элементов неубывающей последовательности элементов, находящихся в файле "temp.dat".
8. За один проход (просмотр файла) вычислить сколько раз в последовательности встречается фрагмент 1, 2, 3, 4, 5, находящийся в файле "temp.dat".
9. За один проход (просмотр файла) вычислить сколько раз в последовательности встречается фрагмент 1, 2, 1, 3, находящийся в файле "temp.dat".
10. Коэффициенты многочлена сведены в последовательность в порядке возрастания степеней, находящуюся в файле "temp.dat". За один проход (просмотр файла) найти значение многочлена и его производной в точке x.
11. Коэффициенты многочлена сведены в последовательность в порядке убывания степеней, находящуюся в файле "temp.dat". За один проход (просмотр файла) найти значение многочлена и его производной в точке x.
12. Дано целое число. Получить число, десятичная запись которого содержит цифры исходного числа в обратном порядке.
13. Вычислить представление числа  $1/n$  в виде десятичной дроби (указать ее начало и период).
14. Возвести число A в степень N, используя не более  $2 * \log(N)$  умножений.
15. Симметрична ли таблица? Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
16. Записать элементы таблицы в обратном порядке: первый элемент должен оказаться на последнем месте, второй - на предпоследнем и т.д. Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
17. Циклически сдвинуть элементы таблицы вправо: первый элемент должен оказаться на втором месте, второй - на третьем и т.д. Последний элемент должен оказаться на первом месте. Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
18. Циклически сдвинуть элементы таблицы из N элементов на k мест влево за время  $O(N)$ . Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
19. Каждый элемент таблицы с индексом от 2 до N-1 заменить на полусумму соседей. Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
20. Известно, что элементы таблицы не убывают. Проведя не более  $1 + \log(N)$  сравнений числа x с элементами таблицы, выяснить, встречается ли x среди элементов таблицы. Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
21. Дано, что первые k-1 элементов таблицы не убывают. Переставить первые k элементов так, чтобы они не убывали. Составить программу, аргументами которой являются длина N линейной таблицы A и сама таблица A[1:N]. Использовать дополнительные таблицы нельзя.
22. Даны k отрезков на прямой. Вычислить, образуют ли они покрытие отрезка [0, 1].
23. Даны k отрезков на прямой. Найти максимальное n, для которого существует точка, принадлежащая одновременно n отрезкам.
24. Известно, что первые N элементов таблицы A возрастают и первые M элементов таблицы B возрастают. Найти число K и построить таблицу C, в которой первые K элементов возрастают, причем число x встречается среди первых K элементов C если и только если оно встречается либо в A либо в B.
25. Известно, что первые N элементов таблицы A возрастают и первые M элементов таблицы B возрастают. Найти число K и построить таблицу C, в которой первые K элементов возрастают, причем число x встречается среди первых K элементов C если и только если оно встречается и в A и в B (пересечение).
26. Элемент называется локальным максимумом, если у него нет соседа, большего, чем он сам. Например, в последовательности из одного элемента - один локальный максимум. В последовательности 3, 3, 1, 0, 2, -1

три локальных максимума: первый, второй и предпоследний элементы. Вычислить число локальных максимумов последовательности.

27. Элемент называется локальным максимумом, если у него нет соседа, большего, чем он сам. Например, в последовательности из одного элемента - один локальный максимум. В последовательности 3, 3, 1, 0, 2, -1 три локальных максимума: первый, второй и предпоследний элементы. Вычислить число локальных максимумов матрицы.
28. Число  $n$  называется периодом последовательности, если любые два элемента последовательности начиная с некоторого, номера которых отличаются на  $n$  совпадают. Отображение целочисленного отрезка  $[1, N]$  в себя задано в виде  $y=f(x)$ . Найти минимальный период последовательности 1,  $f(1)$ ,  $f(f(1))$ , ...
29. Стандартным прямоугольником на плоскости называется множество точек  $(x_1, x_2)$ :  $a_1 \leq x_1 \leq b_1$ ,  $a_2 \leq x_2 \leq b_2$ . Стандартный прямоугольник может вырождаться в вертикальный или горизонтальный отрезок, в точку или в пустое множество. Найти периметр и площадь а) пересечения; б) объединения двух стандартных прямоугольников на плоскости.
30.  $k$ -мерным ящиком называется множество  $(x_1, \dots, x_k)$ :  $a_1 \leq x_1 \leq b_1$ , ...,  $a_k \leq x_k \leq b_k$ . Найти площадь поверхности и объем пересечения и объединения двух  $k$ -мерных ящиков ( $k \geq 1$ ).
31. Вычислить четность произвольной перестановки чисел  $(1, 2, \dots, n)$  за время  $O(n)$ .
32. Представляя комплексные числа парами действительных чисел вида record (первый элемент пары - вещественная, второй - мнимая часть комплексного числа), написать программы сложения, вычитания, умножения, деления двух комплексных чисел.
33. Представляя комплексные числа парами действительных чисел вида record (первый элемент пары - вещественная, второй - мнимая часть комплексного числа), написать программу вычисления модуля, аргумента, корня квадратного числа и сравнения двух комплексных чисел. Проверить на ЭВМ правило сложения аргументов при умножении комплексных чисел.
34. Найти наибольший общий делитель (необязательно положительных) целых чисел, не равных одновременно нулю.
35. Написать программу умножения многочленов  $P$  и  $Q$ .
36. Написать программу, которая записывает в файл "temp.res" первые  $n$  простых чисел.
37. Написать программу, которая записывает в файл "temp.res" разложение на простые множители натурального числа  $n \leq 10000$ .
38. Частным и остатком от деления многочлена  $P$  на многочлен  $Q$  называются такие многочлены  $R$ ,  $S$ , что  $P=Q*S+R$ ,  $\deg(R) < \deg(Q)$ . Написать программу деления с остатком двух многочленов, заданных своими коэффициентами.
39. Записать в файл все возможные разбиения множества  $1..N$  на  $K$  частей.
40. Записать в файл все возможные разбиения числа  $N$  на сумму слагаемых.
41. Записать в файл все последовательности цифр 0,1,2 длины  $n$ , в которые ни одна группа цифр не входит два раза подряд.
42. Записать в файл "temp.res" все перестановки чисел от 1 до  $n$  в лексикографическом порядке.
43. Записать в файл "temp.res" все перестановки чисел от 1 до  $n$  так, чтобы каждая пара соседних перестановок отличалась одной транспозицией.
44. Корабли на поле для морского боя заданы матрицей  $n \times n$  из нулей и единиц. Определить число кораблей
45. Найти объединение трех упорядоченных числовых последовательностей.
46. Найти пересечение трех упорядоченных числовых последовательностей.
47. Дано множество точек на плоскости. Существует ли в этом множестве тройка точек, лежащих на одной прямой?
48. В пространстве дано  $N$  точек с массами  $M_1, \dots, M_N$ . Найти их центр тяжести.
49. Даны декартовы координаты  $N$  ( $N \geq 3$ ) точек плоскости. Отобрать три из них, которые видны из начала координат под наименьшими углами по отношению к вектору  $(X, Y)$ .
50. Выяснить, пересекаются ли два отрезка на плоскости, и если да, то найти точку пересечения.
51. Найти расстояние от точки до отрезка в  $R^2$ ,  $R^3$ .
52. Два треугольника на плоскости заданы координатами своих вершин. Проверить, будут ли они а) равны, б) подобны.
53. Выяснить расположение точки и треугольника в  $R^2$ .
54. Найти вектора биссектрис треугольника в  $R^2$ . Треугольник задается координатами точек.
55. Найти площадь поверхности тетраэдра.
56. Найти минимальный круг, покрывающий множество точек на плоскости.

### Задача 01. Антон решает задачи

Мальчик Антон решает вступительную работу в летний математический лагерь. В ней  $N$  заданий, которые можно выполнять в произвольном порядке. Разные задачи требуют разного времени для решения. При этом известно, что если задание с номером  $i$  выполнять  $j$ -м по счету, Антону потребуется  $T_i * j$  времени: чем больше думаешь, тем больше устаешь. Например, если начать с первой задачи, а затем выполнить вторую, то потребуется  $T_1 * 1 + T_2 * 2$  времени, а если выполнить сначала вторую задачу, а затем первую – то  $T_2 * 1 + T_1 * 2$ . Подскажите Антону, в каком порядке нужно решать задачи, чтобы на выполнение всей работы ушло как можно меньше времени.

#### Входные данные

Вводятся числа  $N, T_1, T_2, \dots, T_N$ , разделенные пробелами. Все числа целые и удовлетворяют следующим ограничениям:  $0 < N \leq 10, 0 < T_i \leq 100$ .

#### Выходные данные

Требуется вывести сначала минимальное время, за которое можно решить все задачи, а затем – номера задач в том порядке, в котором их нужно решать, чтобы уложиться в это время. Все числа разделяются пробелами. Если решений несколько, нужно выдать любое из них.

#### Пример

Входные данные	Выходные данные
2 2 3	7 2 1

### Задача 02. Сережа играет в "Морской бой"

Сережа играет в "Морской бой". Поле для игры представляет собой квадрат  $10 \times 10$  клеток. На поле отмечены клетки, в которые Сережа уже стрелял. Однако пока он не попал ни в один корабль противника. Требуется определить максимальную длину корабля, который может поместиться в небитых клетках этого поля. Корабль представляет из себя прямоугольник ширины 1 и располагается горизонтально или вертикально. (Гарантируется, что на поле есть хотя бы одна небитая клетка.)

#### Входные данные

Вводятся 10 строк по 10 чисел в каждой, числа разделены пробелами. Число 1 означает, что в соответствующую клетку стреляли, число 0 – что в клетку не стреляли.

#### Выходные данные

Требуется вывести одно число от 1 до 10 – максимальную возможную длину корабля.

#### Пример

Входные данные	Выходные данные
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	9

### Задача 03. Сеня наблюдает за кузнечиками

На прямой тропинке на расстоянии 1 метр друг от друга сидят два кузнечика. Время от времени один из кузнечиков прыгает на несколько сантиметров влево или вправо. Требуется узнать, каково было мини-

мальное расстояние, на которое сближались кузнечики в процессе прыжков. (Расстояние считается только в те моменты, когда оба кузнечика сидят на земле).

#### Входные данные

В первой строке вводится одно число  $N$  ( $1 \leq N \leq 100$ ) – общее количество прыжков, а затем  $N$  чисел, описывающих прыжки. Модуль числа равен длине прыжка в сантиметрах; число отрицательное, если кузнечик начинал этот прыжок по направлению к другому кузнечику, и положительное – если *от* другого кузнечика. Числа по модулю не превосходят 100 и все отличны от 0. (Кузнечики могут перепрыгивать друг через друга. Гарантируется, что кузнечики не приземляются друг на друга.)

#### Выходные данные

Требуется вывести одно число – минимальное расстояние в сантиметрах, на которое сближались кузнечики.

#### Пример

Входные данные	Выходные данные
3 –90 10 15	10

#### Задача 04. Боря сортирует матрицу

Назовем таблицу из  $N \times M$  чисел *отсортированной*, если любое число не меньше каждого из чисел, стоящих одновременно выше и левее данного числа (см. пример).

Дана таблица чисел. Требуется переставить числа так, чтобы таблица оказалась отсортированной. Если способов несколько, нужно привести любой из них.

#### Входные данные

Вводятся сначала два числа  $N$  и  $M$  (натуральные, не превосходящие 30), а затем  $N$  строк по  $M$  чисел в каждой, разделенных пробелами. Числа целые и не превышают по модулю 10000.

#### Выходные данные

Вывести  $N$  строк по  $M$  чисел в каждой, разделенных пробелами.

#### Пример

Входные данные	Выходные данные	Комментарий
3 2 1 3 4 9 2 6	3 1 4 6 2 9	Число 6 больше единственного числа, стоящего выше и левее его – числа 3; число 9 больше, чем 3 и 4. Для остальных чисел нет чисел, стоящих выше и левее.

#### Задача 05. Витя и Денис играют в игру "Быки и коровы"

Витя и Денис играли в игру "Быки и коровы". Витя загадал четырёхзначное число с неповторяющимися цифрами, а Денис пытался это число угадать. Для этого он предлагал свои четырёхзначные числа (тоже с неповторяющимися цифрами), а Витя про каждое из них сообщал, сколько в нём "быков" (т. е. цифр, которые не только присутствуют и в Витином числе, и в числе Дениса, но даже стоят на одних и тех же местах) и "коров" (цифр, которые присутствуют в обоих числах, но стоят на разных местах). У них осталась запись партии (последовательность тестовых чисел и ответов на них), но задуманное число утратилось. Восстановите задуманное число.

#### Входные данные



Вводится сначала число  $N$  – количество четырехзначных чисел, названных Денисом в одной партии ( $N \leq 100$ ). Затем вводятся  $N$  строк, по три числа в каждой. Первое – четырехзначное число, названное Денисом (оно не начинается с нуля), второе – количество "быков", третье – количество "коров".

#### Выходные данные

Требуется вывести одно четырехзначное число, задуманное Витей. Это число не начинается с 0. Гарантируется, что ответ в задаче существует и единственный.

#### Пример

Входные данные	Выходные данные
3 1234 0 0 5678 2 2 6578 0 4	5687

#### Задача 06. Ваня наблюдает за лягушкой

Ваня наблюдает за лягушкой. Изначально она сидит в точке 0 числовой прямой. Каждую секунду она прыгает на 1 вправо, пока не достигнет точки  $K$ . Затем она начинает каждую секунду прыгать на 1 влево, пока не вернется в точку 0, затем – опять вправо и т. д. Требуется определить, где окажется лягушка через  $T$  секунд.

#### Входные данные

Вводятся два числа  $K$  и  $T$ , разделенные пробелом. Оба числа натуральные и не превосходят 1 000 000 000.

#### Выходные данные

Вывести одно число – координату лягушки в момент времени  $T$ .

#### Пример

Входные данные	Выходные данные
5 8	2

#### Задача 07. Хождение за золотом

Однажды царь решил вознаградить одного из своих мудрецов за хорошую работу. Он привел его в прямоугольную комнату размером  $N \times M$ , в каждой клетке которой лежало несколько килограммов золота. Царь разрешил мудрецу обойти несколько клеток (переходя с клетки, где сейчас находится мудрец, в одну из четырех с ней соседних), и собрать все золото, которое попадет на его пути. Вам дан маршрут мудреца. Требуется определить, сколько килограммов золота он собрал.

#### Входные данные

Во входном файле записан план комнаты. Сначала записано количество строк  $N$ , затем – количество столбцов  $M$  ( $1 \leq N \leq 20, 1 \leq M \leq 20$ ). Затем записано  $N$  строк по  $M$  чисел в каждой – количество килограммов золота, которое лежит в данной клетке (число от 0 до 50). Далее записано число  $X$  – сколько клеток обошел мудрец. Далее записаны координаты этих клеток (координаты клетки – это два числа: первое определяет номер строки, второе – номер столбца, верхняя левая клетка на плане имеет координаты (1,1), правая нижняя – ( $N, M$ )). Гарантируется, что мудрец не проходил по одной и той же клетке дважды.

#### Выходные данные

В выходной файл выведите количество килограммов золота, которое собрал мудрец.

#### Пример

Входные данные	Выходные данные
3 4	22

1 2 3 4 5 6 7 8 9 10 11 12 5 1 1 2 1 2 2 2 3 1 3	
--	--

### Задача 08. Перевороты

У Пети есть  $N$  кубиков, он выложил их в ряд. После этого пришел Вася и несколько раз переставил в обратном порядке несколько последовательно лежащих кубиков. Чтобы не запутаться, Вася на каждом шаге кроме первого переворачивает только кубики, которые были перевернуты на предыдущем шаге (не обязательно все). Пусть кубики пронумерованы числами от 1 до  $N$  в том порядке, в котором Петя их исходно выложил. Выясните, в каком порядке будут лежать кубики после манипуляций Васи.

Например, пусть у Пети было 10 кубиков, и Вася последовательно переставил в обратном порядке кубики с 1 по 7, со 2 по 5, и с 3 по 5. Тогда кубики окажутся выстроенными в следующем порядке: 7, 6, 2, 5, 4, 3, 1, 8, 9, 10.

#### Формат входных данных

В первой строке файла записано число  $0 \leq N \leq 100000$  количество кубиков, в следующей строке записано число  $m$  – количество переворотов. В следующих  $m$  строках записаны пары чисел  $k_i$  и  $l_i$ , где  $k_i \leq l_i$  и  $k_i \geq k_{i-1}$   $l_i \leq l_{i-1}$

#### Формат выходных данных

В выходной файл записывается полученная последовательность из  $m$  чисел через пробел.

#### Примеры:

Tumble.in	Tumble.out
7 3 3 7 4 6 4 5	1 2 7 5 4 6 3
10 2 3 9 4 4	1 2 9 8 7 6 5 4 3 10
16 6 1 13 1 12 2 11 2 10 3 9 4 8	13 1 11 2 9 4 5 6 7 8 3 10 12 14 15 16

### Задача 09. Спасение царевны

Змей Горыныч похитил царевну Несмеяну. Иванушка Дурачок хочет спасти ее, для этого он добыл Меч-Кладенец. У Змея Горыныча  $N$  голов и  $M$  хвостов. Иванушка может за один удар отрубить не более двух голов или не более двух хвостов. При этом он знает, что если отрубить один хвост, то сразу вырастет два новых, если отрубить сразу два хвоста, то вырастет новая голова, если отрубить голову, то она отрастет заново, а если отрубить две головы, то ничего нового не вырастет. Иванушка хочет отрубить Змею все головы и все хвосты. Помогите ему это сделать. Например, пусть у Горыныча две головы и три хвоста, тогда Иванушке следует отрубить один хвост, после чего их станет четыре, затем по парам срубить хвосты, вырастет еще две головы, теперь можно срубить по парам все четыре головы.

#### Формат входных данных

В первой строке файла записано число  $0 \leq N \leq 1000000$  количество голов у Змея Горыныча.

Во второй строке файла записано число  $0 \leq M \leq 1000000$  количество хвостов у Змея Горыныча.

#### Формат выходных данных

В выходной файл записано количество срубленных одинарных хвостов, затем количество срубленных пар хвостов и количество срубленных пар голов. Если решения нет, то вывести - Нет решения.

#### Примеры:

Dragon.in	Dragon.out
2	1 по 1 хвоста
3	2 по 2 хвоста 2 по 2 головы
6	2 по 2 хвоста
4	4 по 2 головы
3	Нет решения
0	

#### Задача 10. Число ПИ

Одним из методов определения площадей сложных фигур является метод Монте-Карло. Площадь круга этим методом можно определить так:

Вокруг круга описываем квадрат. Далее внутри квадрата наносим случайным образом большое количество точек. Отношение числа точек, попавших в круг, к числу всех точек будет приближённо равно отношению площади круга к площади квадрата.

Вычислить методом Монте-Карло значение числа ПИ для 10, 100, 1000, 10000, 100000 точек.

#### Формат входных данных

нет

#### Формат выходных данных

В выходной файл записано построчно значения числа ПИ для 10, 100, 1000, 10000, 100000 точек

#### Примеры:

NumberPi.in	NumberPi.out
Нет	3,6 3,08 3,076 3,1 3,14136

#### Задача 11. СуперСчастливый билетик

Определить, можно ли между цифрами 6-значного номера автобусного билетика расставить 2 знака «+» или «-» и единственный знак «=» таким образом, чтобы получилось верное равенство. Если да, то вывести все варианты расстановки.

Например:

Ввод: 205632

Вывод: 2-0+5-6=3-2

2-0+5=6+3-2

2=0-5+6+3-2

2+0+5-6=3-2

2+0+5=6+3-2

#### Формат входных данных

В первой строке файла записано шестизначное число  $100\,000 \leq N \leq 999\,999$

#### Формат выходных данных

В выходной файл записаны все возможные комбинации

#### Примеры:

SHappy.in	SHappy.out
205632	2-0+5-6=3-2 2-0+5=6+3-2 2=0-5+6+3-2 2+0+5-6=3-2 2+0+5=6+3-2
123456	Нет решения
211567	2+1+1=5+6-7 2+1+1-5=6-7

#### Задача 12. Найти и заменить

По заказу Ивана Царевича команда программистов под руководством Василисы Премудрой разрабатывает новый пакет программ *Яга Офис*, основой которого будет новый мощный текстовый редактор. Одной из важнейших функций редактора является поиск-замена. Задан текст, набор слов, которые требуется найти в тексте и слова, на которые их следует заменить. Ваша задача - помочь разработчикам редактора реализовать эту сложную функцию.

Например, если исходно был задан текст:

Жили были дед и баба, и было у них три сына.

И набор замен:

дед → Иван

и → да

баба → Марья

сына → гуся

То результатом замен будет:

Жили были Иван да Марья, да было у них три гуся.

#### Формат входных данных

Первая строка файла - исходный текст. Во второй строке записано число  $N$  - количество замен. В следующих  $N$  строках записаны через пробел исходное слово и слово-замена.

#### Формат выходных данных

В выходной файл содержится текст, полученный в результате замены.

#### Примеры:

Replace.in	Replace.out
------------	-------------

Жили были дед и баба, и было у них три сына. 4 дед Иван и да баба Марья сына гуся	Жили были Иван да Марья, да было у них три гуся.

### Задача 13. Странная математика

В связи с реформой образования в тридесятom царстве был введен новый предмет, на котором изучаются различные альтернативные науки. Одной из таких наук является *странная математика*. Ее основное отличие от обычной математики в том, что числа в ней упорядочены не по возрастанию, а лексикографически, то есть как в словаре (сначала по первой цифре, затем, при равной первой цифре - по второй, и так далее). Кроме того, рассматривается не бесконечное множество натуральных чисел, а лишь первые  $N$  чисел. Так, например, если  $N = 11$ , то числа в странной математике оказываются упорядочены следующим образом: 1, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9.

Помогите ученикам в изучении этой науки - напишите программу, которая по заданному  $N$  находит место заданного числа  $X$  в порядке, определенном в странной математике. Например, если  $N = 11$  и  $X = 2$ , то Ваша программа должна выдать в качестве ответа 4.

#### Формат входных данных

В первой строке файла записано число  $1 \leq N \leq 10000$ , во второй – число, которое необходимо найти.

#### Формат выходных данных

В выходной файл записывается номер числа в последовательности.

#### Примеры:

StranMath.in	StranMath.out
11 2	4
22 3	16
101 3	25

### Задача 14. Числа

Даны два целых положительных числа  $A$ ,  $B$ . Напечатать слово "ДА" или "НЕТ" в соответствии с тем, можно ли получить десятичную запись числа  $A$  путем вычеркивания одной или более цифр числа  $B$ .

#### Формат входных данных

В первой строке файла записано число  $1 \leq A \leq 10^9$ , во второй число  $1 \leq B \leq 10^9$ .

#### Формат выходных данных

В выходной файл записано да или нет.

#### Примеры:

Num.in	Num.out
123 13243	да
123	нет

13241	
1234 123	нет

### Задача 15. Фраза

Дано неотрицательное целое число  $K$ , не превышающее миллиона. Напечатать фразу "К ворон" русскими словами. (Пример. Если  $K = 23$ , то должно быть напечатано "двадцать три вороны", если  $K = 3651$  - то "три тысячи шестьсот пятьдесят одна ворона").

#### Формат входных данных

В первой строке файла записано число  $1 \leq N \leq 1000000$

#### Формат выходных данных

В выходной файл записано словесное описание числа.

#### Примеры:

Phrase.in	Phrase.out
11	Одиннадцать ворон
21	Двадцать одна ворона
123	Сто двадцать три вороны

### Задача 16. Прямоугольник

Даны числа  $X_1, Y_1; X_2, Y_2; X_3, Y_3$  - координаты трех каких-то вершин прямоугольника в прямоугольной системе координат. Вычислить и напечатать координаты четвертой вершины.

#### Формат входных данных

Три строки, в каждой из которых через пробел записаны координаты очередной вершины.

#### Формат выходных данных

В выходной файл записаны координаты четвертой вершины прямоугольника.

#### Примеры:

Rectangle.in	Rectangle.out
-2 1 2 1 -2 -1	2 -1
0 0 4 0 0 4	4 4
-2 -1 0 4 3 3	1 -2

### Задача 17. Счастливый билет

Трамвайные билеты имеют номера из шести цифр от 000000 до 999999. Билет называется "счастливым", если сумма 1-й, 3-й и 5-й цифр равна сумме 2-й, 4-й и 6-й цифр. Вычислить и напечатать количество "счастливых" билетов.

#### Формат входных данных

нет

### Формат выходных данных

В выходной файл записано количество счастливых билетов.

### Задача 18. 2007 цифра Фибоначчи

Ряд чисел Фибоначчи представляет собой последовательность натуральных чисел такую, что первое и второе числа равны единице, а каждое следующее равно сумме двух предыдущих:

1 1 2 3 5 8 13 21 34 ...

Числа Фибоначчи выписываются одно за другим вплотную. Определите, какой будет N-ая цифра в такой последовательности. (Обратите внимание, что соответствующее число Фибоначчи может оказаться весьма большим.)

### Формат входных данных

В файл записано число N - номер цифры в последовательности,  $1 \leq N \leq 2006$ .

### Формат выходных данных

В выходной файл записана N-ая цифра

### Примеры:

Fibon.in	Fibon.out
10	55
100	354224848179262000000

### Задача 19. Номера страниц

Любознательный наборщик подсчитал количество литер, потребовавшихся для набора всех номеров страниц в книге. Вы должны по найденному им числу определить, сколько было в этой книге страниц.

### Формат входных данных

В первой строке файла записано N – количество литер  $1 \leq N \leq 10000$

### Формат выходных данных

В выходной файл записано количество страниц.

### Примеры:

NPage.in	NPage.out
8	8
15	12
252	120

### Задача 20. Палиндром

Число называется палиндромом, если его запись читается одинаково сначала и с конца. Для натурального числа n требуется найти все палиндромы состоящие из n цифр квадраты которых тоже являются палиндромами.

### Формат входных данных

В первой строке файла записано количество цифр  $1 \leq N \leq 10$  в числе.

### Формат выходных данных

В выходной файл в каждой строке через пробел записаны число и его квадрат.

### Примеры:

Polindrom.in	Polindrom.out
1	1 1 2 4 3 9
2	11 121 22 484
3	101 10201 111 12321 121 14641 202 40804 212 44944

### Задача 21. Римский счёт

В римской системе счисления используются следующие знаки: I - 1, V - 5, X - 10, L - 50, C - 100, D - 500, M - 1000 соответственно. При записи чисел должны выполняться правила:

- если знак, имеющий меньшее значение следует за знаком, имеющим большее значение, то их значения складываются
- если знак, имеющий меньшее значение предшествует знаку, имеющему большее значение, то их значения вычитаются
- в одной группе не может быть более 3 знаков с одинаковым значением (для единичных разрядов)
- знаки, обозначающие не единичные значения, не могут образовывать групп
- соседними могут быть только знаки, отличающиеся не более чем на порядок в значении

Написать программу перевода числа  $0 < N < 4000$  в римскую систему счисления.

Пример: 3497 = MMMCDXCVII

#### Формат входных данных

В первой строке файла записано число  $1 \leq N \leq 3999$

#### Формат выходных данных

В выходной файл записано его представление в римской системе счисления.

#### Примеры:

Roman.in	Roman.out
10	X
2006	MMVI
3497	MMMCDXCVII

### Задача 22. Перестановка

Запись натуральных чисел 1, 2, ..., n в каком-нибудь порядке называется перестановкой этих чисел. Даны целые числа  $a_1, a_2, \dots, a_n$ . Определить является ли последовательность  $a_1, a_2, \dots, a_n$  перестановкой чисел 1, 2, ..., n?

#### Формат входных данных

В первой строке файла записано количество чисел  $1 \leq N \leq 10000$ , во второй строке - перестановка от 1 до N.

#### Формат выходных данных

В выходной файл записано слово да или нет.

#### Примеры:

Swap.in	Swap.out
5 2 3 1 4 5	Да



7 1 2 7 2 4 5 6	Нет
8 1 1 1 1 1 1 1 1	Нет

### Задача 23. Поезда

Метрополитен города Глупова состоит из единственной однопутной линии. В нулевой момент времени с начальной и конечной станций этой линии навстречу друг другу начинают двигаться два поезда. Их движение подчиняется следующим правилам.

- Отъезжая со станции, поезд сначала разгоняется, потом некоторое (возможно нулевое) время движется с максимальной скоростью, затем замедляется и, в конце концов, останавливается на очередной станции.
- Поезда останавливаются на всех промежуточных станциях метрополитена.
- На каждой из станций поезда стоят одно и тоже фиксированное время.
- Поезда разгоняются и замедляются с одинаковым, постоянным ускорением.
- Поезда имеют одинаковую максимальную скорость.
- Поезда всегда разгоняются до максимальной скорости, если это не мешает остановиться на следующей станции. Иначе они разгоняются, пока это возможно, а затем сразу же начинают тормозить.

Требуется определить, где и когда поезда столкнутся. «Где» определяется расстоянием от начальной станции до места столкновения, «когда» – временем, когда произойдет столкновение.

#### Входные данные

В первой строке входного файла содержится целое число  $N$  ( $2 \leq N \leq 100$ ) – количество станций на линии. Во второй строке записано  $N-1$  вещественное число – расстояние от начальной станции до второй, от начальной до третьей, ..., от начальной до конечной станции. В третьей строке файла записаны три вещественных числа  $A$ ,  $V$ ,  $S$  – ускорение, максимальная скорость и время пребывания поезда на станции соответственно.

#### Выходные данные

В выходной файл вывести расстояние и время с точностью до двух знаков после десятичной точки.

#### Пример

Train.in	Train.out
3 0.25 2.25 1 1 1	0.38 2.50

### Задача 24. Домино

Набор домино состоит из прямоугольных костяшек, каждая из которых разделена на две половинки линией, параллельной более короткой стороне. На каждой из половинок нарисованы точки, количество которых соответствует числу от 0 до  $M$  включительно. На костяшках полного набора домино обозначены все возможные различные пары чисел, например, если  $M$  равно 3, то полный набор содержит 10 костяшек: (0, 0), (0, 1), (0, 2), (0, 3), (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3).

Из костяшек можно выкладывать цепочки, соединяя пары костяшек короткими сторонами, если количества точек на соседних с местом соединения половинках костяшек равны.

Некоторые костяшки были удалены из полного набора. Требуется определить, какое минимальное количество цепочек можно выложить из оставшихся в наборе костяшек, чтобы каждая из них принадлежала ровно одной цепочке.

Напишите программу DOMINO, которая по информации о наборе домино должна ответить, какое минимальное количество цепочек можно выложить.

#### Входные данные

В первой строке входного файла содержится одно целое число  $M$  ( $0 \leq M \leq 100$ ), которое соответствует максимально возможному количеству точек на половинке костяшки. Во второй строке записано одно целое число  $N$ , равное количеству костяшек, удаленных из полного набора. Каждая  $i$ -я из последующих  $N$  строк содержит по два числа  $A_i$  и  $B_i$ . Это количества точек на половинках  $i$ -й удалённой костяшки.

#### Выходные данные

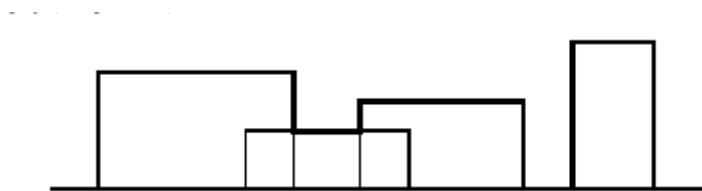
Единственная строка выходного файла DOMINO.SOL должна содержать одно целое число  $L$  - минимальное количество цепочек.

#### Пример

Domino.in	Domino.out
7 2 7 5 3 4	2

#### Задача 25. Прямоугольники

На ось  $OX$  плоскости  $OXY$  положили  $N$  прямоугольников. Требуется найти координаты вершин ломаной, огибающей это множество прямоугольников сверху (см. рис.).



#### Входные данные

Первая строка входного файла содержит целое число  $N$  ( $1 \leq N \leq 100$ ). Далее следуют  $N$  строк, в каждой из которых записана тройка вещественных чисел, описывающих очередной из прямоугольников. Первое из них задает абсциссу левого нижнего угла прямоугольника, а остальные два – его длину и высоту.

#### Выходные данные

В первую строку выходного файла выведите количество вершин искомой ломаной. Далее укажите сами вершины в порядке неубывания абсциссы. Каждая вершина задается своими координатами, записанными через пробел в отдельной строке выходного файла. Никакие два звена ломаной не должны лежать на одной прямой.

#### Пример

Rectangle.in	Rectangle.out
2 0 4 2 2 4 5	6 0 0 0 2 2 2 2 5 6 5 6 0

#### Задача 26. Ограда сада

Где-то в далеком царстве-государстве жил-поживал король. Он унаследовал небольшое собрание редких и весьма ценных деревьев. Для того, чтобы обезопасить свою коллекцию от злоумышленников, король приказал возвести вокруг нее высокий забор. Главный королевский колдун был назначен ответственным за исполнение этого поручения. Увы! Колдун быстро обнаружил, что единственный подходя-

щий материал для постройки забора – это сами деревья. Другими словами, необходимо срубить некоторые деревья для того, чтобы построить забор вокруг оставшихся. Естественно, чтобы сберечь свою голову, колдун захотел минимизировать стоимость срубленных деревьев. Он поднялся в свою башню и оставался там до тех пор, пока не придумал наилучшее возможное решение.

Вы должны написать программу, решающую задачу, с которой столкнулся главный королевский колдун. Постройте такое подмножество деревьев с наименьшей суммарной стоимостью, что, срубив деревья из этого подмножества, можно построить один забор, огораживающий все оставшиеся деревья. Если существует более одного подмножества с минимальной стоимостью, выберите то, в котором меньше деревьев.

### Входные данные

В первой строке входного файла записано целое число  $N$  – количество деревьев в королевском лесу ( $2 \leq N \leq 14$ ). Деревья нумеруются последовательными целыми числами от 1 до  $N$ . Каждая из последующих  $N$  строк содержит четыре целых числа  $x_i, y_i, v_i, l_i$ , описывающих очередное дерево.  $(x_i, y_i)$  – это координаты дерева на плоскости,  $v_i$  – его стоимость, а  $l_i$  – длина забора, который может быть построен из этого дерева. Все числа  $v_i, l_i$ , а также абсолютные величины  $x_i$  и  $y_i$  – целые числа из диапазона  $[0, 10000]$ . Считается, что деревья имеют нулевой радиус.

### Выходные данные

Первая строка выходного файла должна содержать номера деревьев, которые необходимо срубить, разделенные пробелом. Во вторую строку выведите излишек срубленного материала.

### Пример

Ograda.in	Ograda.out
6	2 4 5
0 0 8 3	3.16
1 4 3 2	
2 1 7 1	
4 1 2 3	
3 5 4 6	
2 3 9 8	

## Задача 27. Катаем кубик

На шахматной доске стоит кубик, занимая своим основанием в точности одно из полей доски. На его гранях написаны неотрицательные целые числа, не превосходящие 1000. Кубик можно перемещать на смежные поля, перекатывая через соответствующее ребро в основании. При движении кубика вычисляется сумма чисел, попавших в его основание (каждое число считается столько раз, сколько раз кубик оказывался лежащим на данной грани). Требуется найти такой путь движения кубика между двумя заданными полями доски, при котором вычисленная сумма будет минимальной. Числа, стоящие в основании кубика в начальной и конечной позициях, также входят в сумму.

### Входные данные

Во входном файле через пробел записаны координаты начального и конечного полей и 6 чисел, написанных на передней (в начальный момент), задней, верхней, правой, нижней и левой гранях кубика соответственно. Координаты полей указываются в стандартной шахматной нотации (см. пример). Начальное и конечное поля различны.

### Выходные данные

Выведите в выходной файл минимально возможную сумму и соответствующий ей путь. Путь должен быть задан последовательным перечислением координат полей, по которым движется кубик (включая начальное и конечное поля). Координаты полей записываются в том же формате, что и во входных данных, и разделяются пробелом.

### Пример

Cubic.in	Cubic.out
e2 e3 0 8 1 2 1 1	5 e2 d2 d1 e1 e2 e3

### Задача 28. "ТЕАТР"

В театре  $N$  мест, пронумерованных целыми числами от 1 до  $N$ . Некоторые из зрителей опоздали на спектакль, поэтому после третьего звонка, те зрители, которые имели билеты на неудобные места, пересели на более удобные. Опоздавшие зрители, которые пришли уже после третьего звонка, садились на первое попавшееся свободное место. В антракте один из опоздавших зрителей решил сесть на свое место. Если его место до этого было занято, то тот, кто там сидел, пересаживался на свое место. Если и там кто-то уже сидел, то и этот зритель также вынужден был вернуться на свое место. И так далее. Поскольку в театр попали только зрители, имевшие на руках билеты, то начавшийся в антракте процесс пересаживания зрителей обязательно заканчивался. Необходимо посчитать, сколько человек в результате такого пересаживания были вынуждены поменять на свои места.

Требуется написать программу, которая вычисляет количество зрителей, поменявших свои места из-за опоздания одного зрителя.

#### Формат входных данных:

Входной файл INPUT.TXT состоит из трех строк. В первой строке содержится целое число  $N$  ( $N \leq 30000$ ) — количество мест в зале.

Вторая строка содержит последовательность из  $N$  целых чисел, разделенных пробелами, где первое число определяет номер места в билете у зрителя, который занял место с номером 1, второе - номер места в билете у зрителя, который занял место с номером 2, и так далее. Если место было свободно, то соответствующее число равно 0.

В третьей строке содержится одно число - номер места в билете у опоздавшего зрителя, который в антракте решил пересесть на свое место.

#### Формат выходных данных:

Выходной файл OUTPUT.TXT должен содержать одно число - количество зрителей, поменявших свои места в антракте, включая опоздавшего зрителя.

#### Пример

Theatre.in	Theatre.out
10 0 2 5 3 4 0 0 0 0 0	3

### Задача 29. Цепочка слов

Определить, можно ли построить цепочку из слов, в которой каждое последующее число начинается с той же буквы на которую оканчивается предыдущее. Вывести возможную цепочку, являющуюся решением задачи. Если цепочки нет, вывести ответ "нет".

#### Входные данные

Во входном файле в первой строчке записано количество слов.

Во второй строчке 1-ое слово.

В третьей строчке 2-ое слово.

В  $n+1$  строчке  $n$ -ное слово.

#### Выходные данные

Выведите в выходной файл возможную цепочку или слово «Нет».

#### Пример

Word.in	Word.out
6 ком арбуз лак маска лес ствол	лес, ствол, лак, ком, маска, арбуз.

### Задача 30. Миротворцы

N миротворцев из российского корпуса KFOR десантировались в окрестности аэропорта Слатина. Точка приземления каждого миротворца задается парой целочисленных координат (x, y). За один шаг каждый из десантников может переместиться на соседнюю целочисленную позицию вдоль оси X или Y (т.е. одна из его координат меняется на 1 по абсолютной величине). Шаги делаются по очереди, никакие два миротворца при этом не могут находиться в одной позиции одновременно.

Десантники хотят выстроиться в шеренгу – линию, параллельную одной из осей координат, в которой они стояли бы в подряд идущих целочисленных позициях. Напишите программу, которая определяет минимальное суммарное число шагов, необходимое миротворцам для того, чтобы образовать шеренгу.

#### Входные данные

Первая строка входного файла содержит целое число N – количество миротворцев ( $1 \leq N \leq 10000$ ). Каждая из последующих N строк содержит координаты десантника – два целых числа из диапазона  $[-32768, 32767]$ , разделенные пробелом.

#### Выходные данные

Выведите в выходной файл искомое количество шагов.

#### Пример

Peacemaker.in	Peacemaker.out
3 -1 -1 0 0 1 1	2

### Задача 31. Number

В написанном выражении  $((((1 ? 2) ? 3) ? 4) ? 5) ? 6$  вместо каждого знака ? вставить знак одной из 4 арифметических операций +, -, \*, / так, чтобы результат вычислений равнялся заданному целому числу n (при делении дробная часть отбрасывается).

#### Входные данные

Во входном файле записано одно целое число n, не превосходящее 1000. Найти все решения.

#### Выходные данные

Выведите в выходной файл количество различных решений задачи и сами решения. Если решений нет, вывести 0.

#### Пример

Number.in	Number.out
21	4 (((1+2)+3)+4)+5)+6 (((1*2)+3)*4)-5)+6 (((1*2)-3)+4)*5)+6 (((1*2)*3)+4)+5)+6
28	0

### Задача 32. Лексикографический порядок

Напечатать в лексикографическом порядке все последовательности длины n из чисел в диапазоне от 0 до k-1.

#### Входные данные

Во входном файле в первой строчке записано два числа n и k ( $1 \leq n \leq 10$ ,  $2 \leq k \leq 10$ ,  $n^k \leq 10000$ ).

### Выходные данные

В каждой строке выходного файла вывести  $n$  чисел через пробел - запись соответствующего размещения с повторением.

### Пример

Lexicographic.in	Lexicographic.out
2 2	0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2

### Задача 33. Подмножества

Подмножеством данного множества называют любой набор элементов из данного множества. При этом считается, что все элементы множества различны, и что порядок элементов в подмножестве не имеет значения (то есть  $\{1,3\}$  и  $\{3,1\}$  - это одно и то же подмножество множества  $\{1,2,3\}$ ). Отметим, что у любого множества есть подмножество, в котором нет ни одного элемента:  $\{\}$  (его называют *пустым*), и подмножество, включающее *все* элементы данного множества.

Требуется напечатать все подмножества данного множества  $\{1,2,\dots,n\}$ , исключая пустое.

### Входные данные

Во входном файле записано натуральное число  $n$  -, не превосходящее 10.

### Выходные данные

В каждой строке выходного файла вывести сначала количество чисел в соответствующем подмножестве, а затем сами эти числа. Выводить подмножества можно в любом порядке, в каждом подмножестве числа должны быть упорядочены по возрастанию.

### Пример

Subset.in	Subset.out
2	2 1 2 1 1 1 2

### Задача 34. Последовательности

Напечатайте все последовательности из  $n$  натуральных чисел (возможно, с повторениями), в которых  $i$ -й член не превосходит  $i$ . (на первом месте может стоять только число 1, на втором - 1 или 2, на третьем - 1, 2 или 3, и т.д.). Последовательности требуется вывести в лексикографическом порядке.

### Входные данные

Во входном файле записано натуральное число  $n$ , не превосходящее 8.

### Выходные данные

Выведите в выходной файл В каждой строке вывести  $n$  чисел через пробел - запись соответствующего размещения с повторением

### Пример

SequenceI.in	SequenceI.out
3	1 1 1

	1 1 2
	1 1 3
	1 2 1
	1 2 2
	1 2 3

### Задача 35. Последовательности

Перечислить все возрастающие последовательности длины  $k$  из чисел  $1..n$  в лексикографическом порядке.

#### Входные данные

Во входном файле записано два числа  $n$  и  $k$ .

#### Выходные данные

Выведите в выходной файл все последовательности через пробел.

#### Пример

Sequence.in	Sequence.out
5 2	12 13 14 15 23 24 25 34 35 45

### Задача 36. Убывающие последовательности

Перечислить все убывающие последовательности длины  $k$  из чисел  $1..n$  в лексикографическом порядке.

#### Входные данные

Во входном файле записано два числа  $n$  и  $k$ .

#### Выходные данные

Выведите в выходной файл все последовательности через пробел.

#### Пример

DecreaseSequence.in	DecreaseSequence.out
5 2	21 31 32 41 42 43 51 52 53 54

### Задача 37. Разложение на слагаемые

Напечатать все представления натурального числа  $N$  суммой натуральных чисел. Перестановка слагаемых нового способа не даёт.

#### Входные данные

Во входном файле записано число  $n$ .

#### Выходные данные

В каждой строке выходного файла записано очередное возможное разложение на слагаемые числа  $n$ .

#### Пример

Expansion.in	Expansion.out
4	1+1+1+1 2+1+1 2+2 3+1 4+0

### Задача 38. коды Грея

Перечислить все последовательности длины  $n$  из чисел  $0..1$  в таком порядке, чтобы каждая следующая отличалась от предыдущей в единственной цифре, причём не более чем на 1.

#### Входные данные

Во входном файле записано число  $n$ .

#### Выходные данные

В каждой строке выходного файла вывести очередную последовательность.

#### Пример

CodeGraya.in	CodeGraya.out
3	000 001 011 010 110 111 101 100

### Задача 39. Транспозиция

Напечатать все перестановки чисел  $1..n$  так, чтобы каждая следующая получалась из предыдущей перестановкой (транспозицией) двух соседних чисел.

#### Входные данные

Во входном файле записано число  $n$ .

#### Выходные данные

В каждой строке выходного файла вывести очередную последовательность.

#### Пример

Transposition.in	Transposition.out
3	321 231 213 123 132 312

### Задача 40. Числа Каталана

Перечислить все последовательности длины  $2n$ , составленные из  $n$  единиц и  $n$  минус единиц, у которых сумма любого начального отрезка неотрицательна, т.е. число минус единиц в нём не превосходит числа единиц.

Число таких последовательностей называют *числом Каталана* и оно равно  $C_{2n}^n/(n+1)$ .

#### Входные данные

Во входном файле в первой строчке записано число  $n$ .

#### Выходные данные

В каждой строке выходного файла вывести очередную последовательность.

#### Пример

Katalan.in	Katalan.out
------------	-------------



2	1 -1 1 -1 1 1 -1 -1
---	------------------------

#### Задача 41. Игра Хамелеон

Игра «Хамелеон» происходит в квадрате  $3 \times 3$ , в клетках которого находятся 8 фишек с буквами этого слова, а одна из клеток пуста. За один ход разрешается одну из фишек переместить на соседнюю пустую клетку. Цель игры – достигнуть расположения фишек, указанного на рисунке.

Х	А	М
Е	Л	Е
О	Н	

Напишите программу, которая определяет план достижения цели за минимально возможное число ходов, либо сообщает, что цели достичь нельзя.

#### Входные данные

Во входном файле находится матрица  $3 \times 3$ , составленная из больших букв русского алфавита.

#### Выходные данные

Вывести в первую строку выходного файла искомое число ходов, а в последующие строки – их список. Каждый ход задается координатами той фишки, которая перемещается. Если плана не существует, выведите в выходной файл сообщение «Нет решения».

#### Пример

.in	.out
ХАМ	2
Е Е	3 2
ОЛН	3 3

#### Задача 42. Сплочённая команда

Как показывает опыт, для создания успешной футбольной команды важны не только умения отдельных ее участников, но и сплоченность команды в целом. Характеристикой умения игрока является показатель его профессионализма (ПП). Команда является сплоченной, если ПП каждого из игроков не превосходит суммы ПП любых двух других (в частности, любая команда из одного или двух игроков является сплоченной). Перед тренерским составом была поставлена задача сформировать сплоченную сборную с максимальной суммой ПП игроков (ограничений на количество игроков в команде нет).

Ваша задача состоит в том, чтобы помочь сделать правильный выбор из  $N$  человек, для каждого из которых известен его ПП.

#### Входные данные

В первой строке входного файла записано целое число  $N$  ( $0 \leq N \leq 30000$ ). В последующих  $N$  строках записано по одному целому числу  $P_i$  ( $0 \leq P_i \leq 60000$ ), представляющему собой ПП соответствующего игрока.

#### Выходные данные

В первой строке выходного файла через пробел выведите число игроков, отобранных в команду, и их суммарный ПП. В последующих строках выведите номера игроков, вошедших в команду, в произвольном порядке - по одному числу в строке. Нумерация игроков должна соответствовать порядку перечисления игроков во входном файле. Если ответов несколько, выведите любой из них.

#### Пример

.in	.out
4	3 11
1	2
5	3

3	4
3	

### Задача 43. Треугольник и круг

На плоскости заданы треугольник и круг. Напишите программу определяющую, имеется ли у них хотя бы одна общая точка. И треугольник, и круг считаются включающими свою внутренность, то есть если одна фигура целиком лежит внутри другой, то общая точка есть.

#### Входные данные

Во входном файле в первых трёх строчках записаны координаты вершин треугольника  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$ . В следующей строке записаны координаты центра окружности  $(x, y)$  и её радиус  $r$ .

#### Выходные данные

Выведите в выходной файл слово Да или Нет.

#### Пример

Point.in	Point.out
0 1 1 -1 -1 -1 0 0 1	Да

### Задача 44. Строки в книге

В книге на одной странице помещается  $K$  строк. Таким образом, на 1-й странице печатаются строки с 1-й по  $K$ -ю, на второй — с  $(K+1)$ -й по  $(2 \cdot K)$ -ю и т.д. Напишите программу, которая по номеру строки в тексте определяет номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

#### Входные данные

Входной файл содержит число  $K$  — количество строк, которое печатается на странице, и число  $N$  — номер строки ( $1 \leq K \leq 200$ ,  $1 \leq N \leq 20000$ ).

#### Выходные данные

В выходной файл выведите два числа — номер страницы, на которой будет напечатана эта строка и номер строки на странице

#### Пример

Book.in	Book.out
50 1	1 1
20 25	2 5
15 43	3 13

### Задача 45. Симметричная последовательность

Последовательность чисел назовем *симметричной*, если она одинаково читается как слева направо, так и справа налево. Например, следующие последовательности являются симметричными:

1 2 3 4 5 4 3 2 1  
1 2 1 2 2 1 2 1

Вашей программе будет дана последовательность чисел. Требуется определить, какое минимальное количество и каких чисел надо приписать в конец этой последовательности, чтобы она стала симметричной.

### Входные данные

Во входном файле записано сначала число  $N$  — количество элементов исходной последовательности. Далее записано  $N$  чисел — элементы этой последовательности.  $1 \leq N \leq 100$ , элементы последовательности — натуральные числа от 1 до 9.

### Выходные данные

В выходной файл выведите сначала число  $M$  — минимальное количество элементов, которое надо дописать к последовательности, а потом  $M$  чисел (каждое — от 1 до 9) — числа, которые надо дописать к последовательности.

### Пример

Symmetry.in	Symmetry.out
9 1 2 3 4 5 4 3 2 1	0
5 1 2 1 2 2	3 1 2 1
5 1 2 3 4 5	4 4 3 2 1

## Задача 46. Кинотеатр

$X$  мальчиков и  $Y$  девочек пошли в кинотеатр и купили билеты на подряд идущие места в одном ряду. Напишите программу, которая выдаст, как нужно сесть мальчикам и девочкам, чтобы рядом с каждым мальчиком сидела хотя бы одна девочка, а рядом с каждой девочкой — хотя бы один мальчик.

### Входные данные

Во входном файле записаны два числа —  $X$  и  $Y$  (оба числа натуральные, не превосходящие 100).

Во входном файле в первой строчке записано

### Выходные данные

В выходной файл выведите какую-нибудь строку, в которой будет ровно  $X$  символов  $B$  (обозначающих мальчиков) и  $Y$  символов  $G$  (обозначающих девочек), удовлетворяющую условию задачи. Пробелы между символами выводить не нужно.

Если рассадить мальчиков и девочек согласно условию задачи невозможно, в выходной файл должна быть записана строка NO SOLUTION.

### Пример

Cinema.in	Cinema.out
5 5	BGBGBGBGBG
5 3	BGBGBBGB
100 1	NO SOLUTION

## Задача 47. Количество слов

Во входном файле записана строка текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак,
- символ —, обозначающий в некоторых случаях тире, а в некоторых — дефис.

Слово — это последовательность подряд идущих латинских букв и знаков дефис, ограниченная с обоих концов. В качестве ограничителей могут выступать начало строки, конец строки, пробел, знак

препинания, тире. Тире отличается от дефиса тем, что слева и справа от знака дефис пишутся буквы, а хотя бы с одной стороны от тире идет либо начало строки, либо конец строки, либо пробел, либо какой-либо знак препинания, либо еще одно тире.

Напишите программу, определяющую, сколько слов в данной строке текста.

#### Входные данные

Во входном файле записана строка длиной не более 200 символов.

#### Выходные данные

В выходной файл выведите одно число — количество слов, которые содержатся в исходной строке.

#### Пример

Words.in	Words.out
Hello, world!	2
www.olympiads.ru	3
Gyro-compass – this is a ...	4

### Задача 48. Метро

Метрополитен состоит из нескольких линий метро. Все станции метро в городе пронумерованы натуральными числами от 1 до  $N$ . На каждой линии расположено несколько станций. Если одна и та же станция расположена сразу на нескольких линиях, то она является станцией пересадки и на этой станции можно пересесть с любой линии, которая через нее проходит, на любую другую (опять же проходящую через нее).

Напишите программу, которая по данному вам описанию метрополитена определит, с каким минимальным числом пересадок можно добраться со станции  $A$  на станцию  $B$ . Если данный метрополитен не соединяет все линии в одну систему, то может так получиться, что со станции  $A$  на станцию  $B$  добраться невозможно, в этом случае ваша программа должна это определить.

#### Входные данные

Во входном файле записано сначала число  $N$  — количество станций метро в городе ( $2 \leq N \leq 100$ ). Далее записано число  $M$  — количество линий метро ( $1 \leq M \leq 20$ ). Далее идет описание  $M$  линий. Описание каждой линии состоит из числа  $P_i$  — количество станций на этой линии ( $2 \leq P_i \leq 50$ ) и  $P_i$  чисел, задающих номера станций, через которые проходит линия (ни через какую станцию линия не проходит дважды).

В конце файла записаны два различных: числа  $A$  — номер начальной станции, и  $B$  — номер станции, на которую нам нужно попасть. При этом если через станцию  $A$  проходит несколько линий, то мы можем спуститься на любую из них. Так же если через станцию  $B$  проходит несколько линий, то нам не важно, по какой линии мы приедем.

#### Выходные данные

В выходной файл выведите минимальное количество пересадок, которое нам понадобится. Если добраться со станции  $A$  на станцию  $B$  невозможно, выведите в выходной файл одно число  $-1$  (минус один).

#### Пример

Metro.in	Metro.out
5 2 4 1 2 3 4 2 5 3 3 1	0
5 5 2 1 2 2 1 3 2 2 3 2 3 4 2 4 5	2

1 5	
10 2 6 1 3 5 7 4 9 6 2 4 6 8 10 7 3 8	1
4 2 2 1 2 2 3 4 1 3	-1

#### Задача 49. Метод вращений

Написать программу решения системы линейных алгебраических уравнений путем приведения вещественной матрицы к ступенчатому виду методом вращений.

##### Входные данные

Во входном файле в первой строчке задано  $n$  - порядок матрицы. Следующие  $n$  строк описывают строки расширенной матрицы системы. Числа записываются через пробел.

##### Выходные данные

Выведите в выходной файл построчно все матрицы вращений, промежуточные результаты вычислений и решение системы.

##### Пример

Rotate.in	Rotate.out
3 $\begin{pmatrix} 3 & 1 & 3 \\ 4 & 8 & -6 \\ 12 & 9 & -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ -6 \\ 13 \end{pmatrix}$	$\begin{pmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ -\frac{4}{5} & \frac{3}{5} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 & 3 \\ 4 & 8 & -6 \\ 12 & 9 & -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ -\frac{4}{5} & \frac{3}{5} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 8 \\ -6 \\ 13 \end{pmatrix}$ $\begin{pmatrix} \frac{5}{13} & 0 & \frac{12}{13} \\ 0 & 1 & 0 \\ -\frac{12}{13} & 0 & \frac{5}{13} \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 & 3 \\ 4 & 8 & -6 \\ 12 & 9 & -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{5}{13} & 0 & \frac{12}{13} \\ 0 & 1 & 0 \\ -\frac{12}{13} & 0 & \frac{5}{13} \end{pmatrix} \cdot \begin{pmatrix} 8 \\ -6 \\ 13 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} \\ 0 & \frac{3}{5} & \frac{4}{5} \end{pmatrix} \cdot \begin{pmatrix} 13 & 11 & -3 \\ 0 & 4 & -6 \\ 0 & -3 & 2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} \\ 0 & \frac{3}{5} & \frac{4}{5} \end{pmatrix} \cdot \begin{pmatrix} 12 \\ -10 \\ 5 \end{pmatrix}$ $\begin{pmatrix} 13 & 11 & -3 \\ 0 & 5 & -6 \\ 0 & 0 & -10 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 12 \\ -11 \\ -10 \end{pmatrix}$ $\begin{pmatrix} x = 2 \\ y = -1 \\ z = 1 \end{pmatrix}$

#### Задача 50. Метод Гаусса

Написать программу решения системы линейных алгебраических уравнений путем приведения вещественной матрицы к ступенчатому виду методом Гаусса с выбором максимального по модулю элемента в столбце.

##### Входные данные

Во входном файле в первой строчке задано  $n$  - порядок матрицы. Следующие  $n$  строк описывают строки расширенной матрицы системы. Числа записываются через пробел.

### Выходные данные

Выведите в выходной файл построчно все промежуточные результаты вычислений и решение системы.

### Пример

Gauss.in	Gauss.out
$\begin{pmatrix} 0 & 4 & -6 \\ 5 & 7 & -3 \\ 12 & 9 & -2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -10 \\ 0 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 12 & 9 & -2 \\ 5 & 7 & -3 \\ 0 & 4 & -6 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 13 \\ 0 \\ -10 \end{pmatrix}$ $\begin{pmatrix} 1 & 9/12 & -2/12 \\ 0 & 39/12 & -26/12 \\ 0 & 4 & -6 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 13/12 \\ -65/12 \\ -10 \end{pmatrix}$ $\begin{pmatrix} 1 & 9/12 & -2/12 \\ 0 & 1 & -2/3 \\ 0 & 0 & -10/3 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 13/12 \\ -5/3 \\ -10/3 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$

### Задача 51. Многочлены

Частным и остатком от деления многочлена P на многочлен Q называются такие многочлены R, S, что  $P=Q*S+R$ ,  $\deg(R) < \deg(Q)$ .

Написать программу деления с остатком двух многочленов, заданных своими коэффициентами.

Например:  $(4x^5-3x^2+2x-5)/(x^2+2)=(4x^3-8x-3)*(x^2+2)+(18x+1)$

### Входные данные

Во входном файле в первой строчке записано число n – наибольшая степень многочлена. В следующей строке записаны n+1 число через пробел - коэффициенты многочлена начиная со свободного члена.

### Выходные данные

Выведите в выходной файл

### Пример

Multinomial.in	Multinomial.out
5 -5 2 -3 0 0 4	-3 -8 0 4 1 18

### Задача 52. Римские цифры

Перевести число, введенное римскими цифрами в его арабский эквивалент.

### Входные данные

Во входном файле записано число в римской системе счисления.

### Выходные данные

Выведите в выходной файл это же число в арабском эквиваленте.

### Пример

Rome.in	Rome.out
MMVII	2007

### Задача 53. Комплексная арифметика

Написать программу нахождения комплексных корней квадратного уравнения с комплексными коэффициентами.

Например:  $(2i)x^2 + (-4+2i)x + (-22-6i) = 0$ ;  $x_1 = 2+3i$ ;  $x_2 = -3+i$

#### Входные данные

Во входном файле в строчках записаны пары чисел – коэффициенты квадратного уравнения, начиная со свободного члена.

#### Выходные данные

Выведите в выходной файл в двух строках пары чисел – корни уравнения

#### Пример

Complex.in	Complex.out
-22 -6	2 -3
-4 2	-3 1
0 2	

### Задача 54. Матрица

Вам дана верхнетреугольная квадратная матрица  $A$  размера  $n \times n$ , содержащая только 0, 1 и 2. Будем выполнять все арифметические действия (сложение, вычитание и умножение) по модулю 3. Ваша задача — найти такую матрицу  $B$ , что  $B^2 = A$ .

Напомним, что при умножении двух матриц размера  $n \times n$ , получается матрица такого же размера, причем если  $C = AB$ , то  $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$ .

#### Входные данные

Первая строка входного файла содержит число  $n$  ( $1 \leq n \leq 300$ ). В следующих  $n$  строках располагается матрица  $A$ . Каждый элемент матрицы – целое число от 0 до 2. Гарантируется, что матрица верхнетреугольная, то есть  $A_{ij} = 0$ , если  $i > j$ ;  $i, j = 1, 2, \dots, n$ . Также гарантируется, что на главной диагонали матрицы стоят единицы, то есть для всех  $i$  выполняется равенство  $A_{ii} = 1$ .

#### Выходные данные

Выведите  $n$  строк по  $n$  чисел — матрицу  $B$ . Если искомой матрицы не существует, выведите одно число 0.

#### Пример

Matrix.in	Matrix.out
1	1
1	
2	1 1
1 2	0 1
0 1	

### Задача 55. Строчки

Рассмотрим множество строк, состоящих из маленьких латинских букв. Будем говорить, что строка  $a$  длины  $l$  входит в строку  $b$  как подстрока, если существует такое  $k$ , что  $b[k] = a[1]$ ,  $b[k+1] = a[2]$ , ...,  $b[k+l-1] = a[l]$ . Количеством вхождений строки  $a$  в строку  $b$  назовем количество различных индексов  $k$ , удовлетворяющих приведенному условию.

Вам дана строка  $s$ . Подсчитайте, сколько существует различных строк  $w$  длины  $m$ , таких, что  $s$  входит в  $w$  ровно  $n$  раз.

#### Входные данные

Первая строка входного файла содержит  $s$ . Длина  $s$  не превосходит 20. Вторая строка содержит числа  $m$  и  $n$  ( $1 \leq m \leq 100$ ,  $0 \leq n \leq 20$ ).

#### Выходные данные

В единственную строку выходного файла выведите целое число без лидирующих нулей — ответ на поставленную задачу.

#### Пример

Substring.in	Substring.out
aba 6 2	53

#### Задача 56. Реклама

В супермаркете решили время от времени транслировать рекламу новых товаров. Для того, чтобы составить оптимальное расписание трансляции рекламы, руководство супермаркета провело следующее исследование: в течение дня для каждого покупателя, посетившего супермаркет, было зафиксировано время, когда он пришел в супермаркет, и когда он из него ушел.

Менеджер по рекламе предположил, что такое расписание прихода-ухода покупателей сохранится и в последующие дни. Он хочет составить расписание трансляции рекламных роликов, чтобы каждый покупатель услышал не меньше двух рекламных объявлений. В то же время он выдвинул условие, чтобы два рекламных объявления не транслировались одновременно и, поскольку продавцам все время приходится выслушивать эту рекламу, общее число рекламных объявлений за день было минимальным.

Напишите программу, которая составит такое расписание трансляции рекламных роликов. Рекламные объявления можно начинать транслировать только в целые моменты времени. Считается, что каждое рекламное объявление заканчивается до наступления следующего целого момента времени. Если рекламное объявление транслируется в тот момент времени, когда покупатель входит в супермаркет или уходит из него, покупатель это объявление услышать успевает.

#### Входные данные

Во входном файле записано сначала число  $N$  — количество покупателей, посетивших супермаркет за день ( $1 \leq N \leq 3000$ ). Затем идет  $N$  пар натуральных чисел  $A_i, B_i$ , задающих соответственно время прихода и время ухода покупателей из супермаркета ( $0 < A_i < B_i < 10^6$ ).

#### Выходные данные

В выходной файл выведите сначала количество рекламных объявлений, которое будет протранслировано за день. Затем выведите в возрастающем порядке моменты времени, в которые нужно транслировать рекламные объявления.

Если решений несколько, выведите любое из них.

#### Пример

Reclama.in	Reclama.out
5 1 10 10 12 1 10 1 10 23 24	5 5 10 12 23 24

#### Задача 57 Юный поджигатель

На клеточном поле введена система координат так, что центр координат находится в точке пересечения линий сетки и оси направлены вдоль линий сетки.

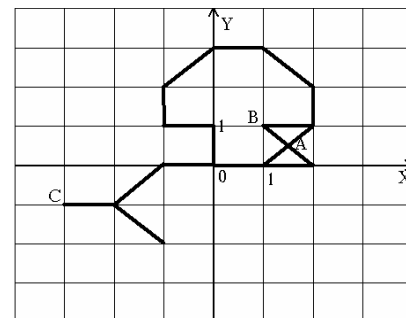
На этом поле выложили связную фигуру, состоящую из спичек. Использовались спички двух типов:



- Спички длины 1 выкладывались по сторонам клеток.
- Спички длины  $\sqrt{2}$  выкладывались по диагоналям клеток.

Ребенок хочет сжечь фигуру. При этом он может поджечь ее в одной точке, имеющей целочисленные координаты (например, в точке А на рисунке поджигать фигуру нельзя, а в точках В и С — можно).

Известно, что огонь распространяется вдоль спички равномерно (но по каждой спичке — со своей скоростью). Спичка может гореть в нескольких местах (например, когда она загорается с двух концов; или когда в середине диагональной спички огонь перекидывается с одной спички на другую — огонь расплзается по вновь подожженной спичке в обе стороны).



Напишите программу, которая определит, в какой точке нужно поджечь фигуру, чтобы она сгорела за минимальное время.

### Входные данные

Во входном файле записано сначала число  $N$  — количество спичек ( $1 \leq N \leq 40$ ). Затем идет  $N$  пятерок чисел вида  $X_1, Y_1, X_2, Y_2, T$ , задающих координаты концов спички и время ее сгорания при условии, что она будет подожжена с одного конца (гарантируется, что каждая спичка имеет длину 1 или  $\sqrt{2}$ , все спички образуют связную фигуру, и положение никаких двух спичек не совпадает). Все координаты — целые числа, по модулю не превышающие 200, время сгорания — натуральное число, не превышающее  $10^7$ .

### Выходные данные

Выведите координаты целочисленной точки, в которой нужно поджечь фигуру, чтобы она сгорела за наименьшее время, а затем время, за которое в этом случае фигура сгорит. Время должно быть выведено с точностью не менее 2-х знаков после десятичной точки. Если решений несколько, выведите любое из них.

### Пример

Flame.in	Flame.out
1 0 0 1 1 1	0 0 1.00
5 0 0 0 1 1 1 0 0 1 10 0 0 1 0 1 0 0 1 1 1 2 2 1 1 1	0 0 3.25
3 1 1 1 2 10 1 2 2 2 10 1 1 2 2 50	2 2 35.00
16 0 0 0 1 1    -2 -1 -3 -1 1    -2 -1 - 1 0 1    -2 -1 -1 -2 1 -1 0 0 0 1    0 3 1 3 1    1 3 2 2 1    2 2 2 1 1 2 1 1 0 1    2 0 1 1 1    2 0 1 0 1 2 1 1 1 1 0 0 1 0 1    0 1 -1 1 1    -1 1 -1 2 1    0 3 -1 2 1	0 0 4.50

### Задача 58. Поле чудес

Для игры в "Поле чудес" используется круглый барабан, разделенный на сектора, и стрелка. В каждом секторе записано некоторое число. В различных секторах может быть записано одно и то же число.

Однажды ведущий решил изменить правила игры. Он сам стал вращать барабан и называть игроку (который барабана не видел) все числа подряд в том порядке, в котором на них указывала стрелка в процессе вращения барабана. Получилось так, что барабан сделал целое число оборотов, то есть последний сектор совпал с первым.

После этого ведущий задал участнику вопрос: какое наименьшее число секторов может быть на барабанае? Напишите программу, отвечающую на этот вопрос.

#### Входные данные

Во входном файле записано сначала число  $N$  — количество чисел, которое назвал ведущий ( $2 \leq N \leq 30000$ ). Затем записано  $N$  чисел, на которые указывала стрелка в процессе вращения барабана. Первое число всегда совпадает с последним (в конце стрелка указывает на тот же сектор, что и в начале). Числа, записанные в секторах барабана, — натуральные, не превышающие 32000.

#### Выходные данные

Выведите минимальное число секторов, которое может быть на барабанае.

#### Пример

FieldMiracles.in	FieldMiracles.out
13 5 3 1 3 5 2 5 3 1 3 5 2 5	6
4 1 1 1 1	1
4 1 2 3 1	3

#### Задача 59. Квадрат

Требуется в каждую клетку квадратной таблицы размером  $N \times N$  поставить ноль или единицу так, чтобы в любом квадрате размера  $K \times K$  было ровно  $S$  единиц.

#### Входные данные

Во входном файле записаны три числа —  $N, K, S$  ( $1 \leq N \leq 100, 1 \leq K \leq N, 0 \leq S \leq K^2$ ).

#### Выходные данные

В выходной файл выведите заполненную таблицу. Числа в строке должны разделяться пробелом, каждая строка таблицы должна быть выведена на отдельной строке файла. Если решений несколько, выведите любое из них.

#### Пример

Square.in	Square.out
3 2 1	0 0 0 0 1 0 0 0 0
4 2 2	1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0

#### Задача 60. Последовательность

Дана последовательность целых чисел  $a_1, a_2, \dots, a_n$ , ( $1 \leq n \leq 100000$ ) каждое из которых по модулю не превосходит 10000. Эта последовательность записана на бумажной ленте, которая свернута в кольцо. Разрежем эту кольцо в некоторой точке между числами, получим полоску с записанной на ней последовательностью следующего вида:

$a_j, a_{j+1}, \dots, a_n, a_1, a_2, \dots, a_{j-1}$ .

Назовем точку разреза *хорошей*, если все частичные суммы полученной последовательности строго положительны:

$$a_j > 0,$$

$$a_j + a_{j+1} > 0,$$

.....

$$a_j + a_{j+1} + \dots + a_n > 0,$$

$$a_j + a_{j+1} + \dots + a_n + a_1 > 0,$$

.....

$$a_j + a_{j+1} + \dots + a_n + a_1 + a_2 + \dots + a_{j-2} > 0,$$

$$a_j + a_{j+1} + \dots + a_n + a_1 + a_2 + \dots + a_{j-2} + a_{j-1} > 0.$$

Вам требуется вычислить число хороших точек разреза.

### Входные данные

Первая строка входного файла содержит число  $n$ , а во второй строке заданы числа  $a_1, a_2, \dots, a_n$ .

### Выходные данные

Выведите в выходной файл число искомых хороших точек разреза.

### Пример

Sequence.in	Sequence.out
5 0 1 -2 10 3	2