

ЕГЭ по информатике. PASCAL

Язык программирования: PASCAL

Сергей Митрофанов

E-mail: infostar@mail.ru

<http://www.Best-Listing.ru/>

В сборнике 73 задачи

Гимназия "Лаборатория Салахова"

Сургут

10 апреля 2016 года

Содержание

1	Задачи С2 [с2_11]	3
2	Задачи С4 [с4_20]	7
3	Основные алгоритмы [alg_40]	23
4	Задача 25, ЕГЭ-2016 [z25_2]	27

1 Задачи C2 [c2_11]

- c2_1.** На вход программы подаются строчные английские буквы. Ввод этих символов заканчивается точкой (другие символы, отличные от точки и букв 'a'..'z', во входных данных отсутствуют). Требуется написать программу, которая будет печатать буквы, встречающиеся во входной последовательности, в порядке уменьшения частоты их встречаемости. Каждая буква должна быть распечатана один раз. Точка при этом не учитывается. Если какие-то буквы встречаются одинаковое число раз, то они выводятся в алфавитном порядке. Например, пусть на вход подаются следующие символы: *batat*. В данном случае программа должна вывести: *atb*.
- c2_2.** Опишите на русском языке или одном из языков программирования алгоритм получения из заданного целочисленного массива размером 30 элементов другого массива, который будет содержать модули значений элементов первого массива (не используя специальной функции, вычисляющей модуль числа).
- c2_3.** Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют четное значение и не делятся на три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого четно и не кратно трем.
- c2_4.** Дан целочисленный массив из 31 элемента, содержащий среднесуточную температуру в градусах Цельсия в городе N для каждого дня в марте. Опишите алгоритм, который находит и выводит самую высокую температуру за все воскресения марта, если 1-го марта была пятница. Известно, что температура не опускалась ниже -30 С. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

```
Const N = 31;
```

```
Var
```

```
  A : array [1..N] of Integer;
```

```
  i, j, m : Integer;
```

```
Begin
```

```
for i := 1 to N do
  ReadLn (A [i]);
```

c2_5. На пустой шахматной доске в одной из клеток стоит шахматный конь. Опишите на русском языке или одном из языков программирования алгоритм получения списка клеток, которые конь может достичь за один ход из данной клетки. На вход программы поступают два целых числа: x, y ($1 \leq x, y \leq 8$) – координаты клетки, в которой стоит конь. На выходе программы должен быть выведен список пар целых чисел – координаты клеток, достижимых конем за один ход.

c2_6. Автобусный билет имеет номер, состоящий из 6 цифр. Билет считается счастливым, если сумма первых 3 цифр равна сумме 3 последних цифр. Например, 143 503 или 001 100. Любое сочетание из 6 цифр считается допустимым для номера билета. Номера, состоящие из шести 0 (000 000) и шести 9 (999 999), также допустимы.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая должна определять количество всех счастливых билетов.

Программа должна выводить только 1 число – количество счастливых билетов.

Найти количество счастливых билетов от 000000 до 999999. Представить самый эффективный алгоритм.

c2_7. Банковский счет с ежемесячной выплатой процентов и капитализацией предусматривает увеличение суммы банковского счета на сумму, составляющую процент от суммы, хранящейся на счете. При этом полученная прибавка суммы счета суммируется с общим количеством денег на счете и используется при расчете прибавки на следующий месяц. Задан годовой процент p , больший ежемесячного процента в 12 раз. Определить, какой должна быть начальная сумма вклада x для получения через m полных месяцев суммы y . Результат вычислений округлить согласно правилам арифметического округления до второго знака после запятой.

Формат входного файла

Во входном файле input.txt находятся вещественные числа y ($1 \leq y \leq 100000$), p ($1 \leq p \leq 100$) и целое число m ($1 \leq m \leq 1200$). Точность представления вещественных чисел – до второго знака

после запятой.

Формат выходного файла

В выходной файл *output.txt* вывести величину x . Точность представления до второго знака после запятой.

Пример входных и выходных данных

input.txt	output.txt
12000.00 12.00 12	10649.39

c2_8. Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), на вход которой подается натуральное число N . Программа должна разложить это число на простые множители. На выходе должно быть распечатано разложение числа, в котором указано произведение простых сомножителей с указанием их степеней (без повторений самих сомножителей).

Например, для $N = 2000$ верным выводом является строка $2^4 * 5^3$ и неверным выводом - строка: $2 * 2 * 2 * 2 * 5 * 5 * 5$.

c2_9. Одномерный массив содержит 10000 неотрицательных целых чисел, на хранение каждого из которых отводится один байт. Составьте наиболее эффективную программу вывода на экран упорядоченных по возрастанию элементов такого массива.

c2_10. Пусть F, G – логические функции от двух логических переменных, заданные своими столбцами таблицы истинности. Составьте программу, которая по введенным столбцам таблицы истинности функций F, G выводит столбец истинности функции $(x, y, z) = G(F(x, y), z)$.

c2_11. Дан массив, содержащий неотрицательные целые числа, не превышающие 10 000.

Необходимо вывести:

- минимальный четный элемент, если количество четных элементов не больше, чем нечетных;
- минимальный нечетный элемент, если количество нечетных элементов меньше, чем четных.

Например, для массива из шести элементов, равных соответственно 4, 6, 12, 17, 9, 8, ответом будет 9 – наименьшее нечетное число, поскольку нечетных чисел в этом массиве меньше.

Напишите на одном из языков программирования программу для

решения этой задачи. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль

```
Const N = 2000;
```

```
Var
```

```
  A : array [1..N] of Integer;
```

```
  i, j, k, m : Integer;
```

```
Begin
```

```
  for i := 1 to N do
```

```
    ReadLn (A [i]);
```

```
  ...
```

```
End.
```

2 Задачи С4 [с4_20]

с4_1.

С4, 2008.

На вход программе подаются сведения об ячейках автоматической камеры хранения багажа. В первой строке задана текущая дата: через точку два целых числа, соответствующие дню (от 01 до 31 – ровно 2 символа) и месяцу (от 01 до 12 – ровно 2 символа). Во второй строке сообщается количество занятых ячеек N , которое не меньше 3, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат: <номер ячейки> <дата сдачи багажа>, где <номер ячейки> – четырехзначное число, <дата сдачи багажа> – через точку два целых числа, соответствующие дню (от 01 до 31 – ровно 2 символа) и месяцу (от 01 до 12 – ровно 2 символа). Номер ячейки и дата сдачи багажа разделены одним пробелом. Сведения отсортированы в порядке номеров ячеек.

Время хранения багажа – трое суток. Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal v. 7.0), которая выведет номера ячеек, в которых багаж хранится заведомо больше трех суток – то есть разница между датой сдачи багажа и текущей датой составляет 4 и более дней. Номера ячеек следует выводить в хронологическом порядке сдачи багажа.

Багаж мог сдаваться только в текущем или предыдущем месяце текущего календарного года (если текущий месяц январь, то данные о сдаче багажа в декабре прошлого года отсутствуют). Количество дней в каждом из месяцев текущего года следующее: январь – 31, февраль – 28, март – 31, апрель – 30, май – 31, июнь – 30, июль – 31, август – 31, сентябрь – 30, октябрь – 31, ноябрь – 30, декабрь – 31. Все входные данные корректны.

Пример входных данных:

```
04.06
3
1000 01.06
1001 31.05
2007 21.05
```

Результат работы программы для этого примера:

```
2007
1001
```

с4_2.

На вход программе подаются сведения о номерах школ бального танца, воспитанники которых принимают участие в соревнованиях. В первой строке сообщается количество пар N каждая из следующих N строк имеет формат: $\langle \text{Фамилия1} \rangle \langle \text{Фамилия2} \rangle \langle \text{номер школы} \rangle$, где $\langle \text{Фамилия} \rangle$ – строка, состоящие не более чем из 20 символов, $\langle \text{Номер школы} \rangle$ – не более чем двузначный номер. $\langle \text{Фамилия1} \rangle$ и $\langle \text{Фамилия2} \rangle$, а также $\langle \text{Фамилия2} \rangle$ и $\langle \text{Номер школы} \rangle$ разделены одним пробелом.

Пример входной строки:

Иванова Петров 57

Требуется написать программу (с указанием используемой версии языка программирования, например Borland Pascal 7.0), которая будет выводить на экран информацию, из каких школ было меньше всего участников соревнования (но из этих школ была хотя бы одна пара).

с4_3.

Вступительные испытания в некоторый ВУЗ состоят из трех экзаменов: математика (максимальный балл – 9), информатика (максимальный балл – 9), литература (максимальный балл – 5). На вход программе подаются сведения о сдаче этих экзаменов абитуриентами. В первой строке вводится количество абитуриентов N , во второй – количество мест K ($K < N$) на которые эти абитуриенты претендуют. Каждая из следующих N строк имеет следующий формат: $\langle \text{Фамилия} \rangle \langle \text{оценка1} \rangle \langle \text{оценка2} \rangle \langle \text{оценка3} \rangle$, где $\langle \text{Фамилия} \rangle$ – строка, состоящая не более, чем из 20 символов, оценки – числа от 0 до максимальной оценки по предмету соответственно. (Ноль ставится в случае, если экзамен не сдавался, например, после полученной на предыдущем экзамене двойки. Все баллы, большие 2, считаются удовлетворительными).

Пример входных строк:

Иванов 8 9 3

Петров 2 0 0

Сидоров 3 2 0

Требуется написать программу, которая определяла бы по имеющимся данным количество абитуриентов, набравших полупроходной балл в данный ВУЗ или сообщала, что такой балл отсутствует. (Полупроходным называется такой балл, что лишь часть абитуриентов, набравших такой балл и не получивших ни одной неудовлетворительной оценки, попадает в K лучших, которые должны быть зачислены на 1 курс). Считается, что абитури-

енты, получившие только удовлетворительные удовлетворительные оценки, обязательно присутствуют.

Примечание о полупроходном балле.

Пример 1.

Пусть $N = 8$, $K = 4$ и

Иванов 9 9 5
Петров 9 9 4
Сидоров 5 5 5
Скворцов 5 5 5
Кисин 5 5 4
Наумов 5 5 4
Абросимов 3 3 3
Катасонов 3 2 0

В данном случае полупроходной балл отсутствует.

Пример 2.

Пусть $N = 8$, $K = 4$ и

Иванов 9 9 5
Петров 9 9 4
Сидоров 5 5 5
Скворцов 5 5 5
Кисин 5 5 5
Наумов 5 5 5
Абросимов 5 5 5
Катасонов 3 3 3

Полупроходной балл присутствует, и он равен 15. Количество абитуриентов, набравших полупроходной балл равно 5.

с4_4.

Заключительный этап олимпиады по физике проводился для учеников 9-11-х классов. Ученики 7-8-х классов выступали за девятый класс. Каждый участник олимпиады мог набрать от 0 до 100 баллов. По положению победителями олимпиады признаются участники, набравшие баллов больше, чем другие участники той же параллели. Если больше 45% участников из одной параллели получили равный максимальный балл, то все они признаются победителями в случае, если они набрали больше половины возможных баллов, в противном случае, победителей в данной параллели нет.

Напишите эффективную по времени исполнения и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет

определять для каждой параллели, сколько человек в ней являются победителями олимпиады. На вход программе сначала подается число участников олимпиады N . В каждой из следующих N строк находится результат одного из участников олимпиады в следующем формате: <Фамилия> <Имя> <класс> <баллы> где <Фамилия> – строка, состоящая не более, чем из 20 символов, <Имя> – строка, состоящая не более, чем из 15 символов, <класс> – число от 7 до 11, <баллы> – целое число от 0 до 100 набранных участником баллов. <Фамилия> и <Имя>, <Имя> и <класс>, а также <класс> и <баллы> разделены одним пробелом.

Пример входной строки: Иванов Петр 10 75

Программа должна выводить через пробел 3 числа - количество победителей среди 9-, 10- и 11-классников соответственно. Если в какой-то параллели победителей нет, то вместо соответствующего числа выводится 0. Гарантируется что в каждой параллели есть хотя бы один участник с ненулевыми баллами.

Пример выходных данных:

0 2 1

c4_5.

На вход программе из текстового файла подается последовательность символов, представляющая собой квадратное уравнение (первая строка заданного файла). Гарантируется, что уравнение имеет "хороший вид", а все коэффициенты определены.

При этом условлено, что при вводе

$$x^2 = a$$

$$x = b$$

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая дописывает в конец файла корни заданного квадратного уравнения.

Например, пусть на вход в программу из файла подаются следующие данные:

$$2a-4b-6$$

В данном случае программа должна дописать в файл:

-1

3

c4_5v2.

На вход программе из текстового файла подается последовательность символов, представляющая собой квадратное уравнение

(первая строка заданного файла). Гарантируется, что уравнение имеет "хороший вид", а все коэффициенты определены.

При этом условлено, что при вводе

$$x^2 = a$$

$$x = b$$

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая дописывает в конец файла корни заданного квадратного уравнения.

Например, пусть на вход в программу из файла подаются следующие данные:

$$2a-4b-6$$

В данном случае программа должна дописать в файл:

-1

3

c4_6.

На вход программе подаются сведения о пассажирах, сдавших свой багаж в камеру хранения. В первой строке задано текущее время: через двоеточие два целых числа, соответствующие часам (от 00 до 21 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа). Во второй строке сообщается количество пассажиров N , которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат: <Фамилия> <время освобождения ячейки>, где <Фамилия> – строка, состоящая не более, чем из 20 символов, <время освобождения ячейки> разделены одним пробелом. Сведения отсортированы в порядке времени сдачи багажа.

Требуется написать программу, выводящую фамилии пассажиров, которые в ближайшие 2 часа должны освободить ячейки, в хронологическом порядке освобождения ячеек.

Пример входных данных:

10:00

3

Иванов 12:00

Петров 10:00

Сидоров 12:12

Результат работы программы для этих данных:

Петров

Иванов

с4_7.

Во входном файле `meteo.dat` 365 строк, которые содержат информацию о среднесуточной температуре всех дней 2003 г. Формат каждой из строк следующий: сначала записана дата в виде `dd.mm` (на запись номера дня и номера месяца в числовом формате отводится строго два символа, день от месяца отделен точкой), затем через пробел записано значение температуры – число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, то есть хронологический порядок нарушен. Требуется написать программу, которая рассчитает и выведет на экран следующую выходную информацию:

- 1) среднегодовую температуру (в отдельной строке вывода);
- 2) информацию о месяцах, для которых абсолютная величина разности среднемесячной и среднегодовой температуры не превосходит 5 градусов. Значения для каждого из месяцев, удовлетворяющих этому условию, следует выводить в отдельной строке в виде: номер месяца, значение среднемесячной температуры.

с4_8.

На вход программы подаются фамилии и имена учеников. Известно, что общее количество учеников не превосходит 100. В первой строке вводится количество учеников, принимавших участие в соревнованиях, N . Далее следуют N строк, имеющих следующий формат: `<Фамилия> <Имя>`. Здесь `<Фамилия>` – строка, состоящая не более чем из 20 символов; `<Имя>` – строка, состоящая не более чем из 15 символов; при этом `<Фамилия>` и `<Имя>` разделены одним пробелом.

Примеры входных строк:

Иванова Мария

Петров Сергей

Требуется написать программу, которая формирует и печатает уникальный логин для каждого ученика по следующему правилу: если фамилия встречается первый раз, то логин – это данная фамилия, если фамилия встречается второй раз, то логин – это фамилия, в конец которой приписывается число 2 и т.д.

Например, для входной последовательности

Иванова Мария

Петров Сергей

Бойцова Екатерина

Петров Иван

Иванова Наташа

будут сформированы следующие логины:

Иванова
Петров
Бойцова
Петров2
Иванова2

с4_9.

По каналу связи передается последовательность положительных целых чисел, все числа не превышают 1000, их количество заранее неизвестно. Каждое число передается в виде отдельной текстовой строки, содержащей десятичную запись числа. Признаком конца передаваемой последовательности является число 0.

Напишите эффективную программу, которая вычисляет длину наибольшего фрагмента идущих подряд элементов последовательности, в котором каждое следующее число меньше предыдущего. Программа должна напечатать отчет по следующей форме: Получено ... чисел Длина максимального участка убывания: ...

Пример входных данных

144
17
21
19
19
76
70
55
30
100
70
0

Пример выходных данных для приведенного выше примера входных данных

Получено 11 чисел
Длина максимального участка убывания: 4

с4_10.

С4, демонстрационный вариант 2009.

На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат: <Фамилия> <Инициалы> <номер школы>, где <Фамилия> –

строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> – не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:

Иванов П.С. 57

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из какой школы было меньше всего участников (таких школ может быть несколько). При этом необходимо вывести информацию только по школам, пославшим хотя бы одного участника, Следует учитывать, что $N \geq 1000$. (4 балла)

c4_11.

На вход программе подаются сведения о номерах регионов России, приславших свои команды для участия в автомобильных гонках.

В первой строке сообщается количество команд N .

Каждая из следующих N строк имеет формат: <Фамилия> <Инициалы> <номер региона>, где <Фамилия> – строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4 символов (буква, точка, буква, точка), <Номер региона> – не более чем из двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <Номер региона> разделены одним пробелом.

Пример входной строки:

Козлов П.С. 56

Требуется написать программу (с указанием используемой версии языка программирования, например Borland Pascal 7.0), которая будет выводить на экран сведения, из каких регионов (или региона) было меньше всего участников гонок (даже если из этих регионов был хотя бы один участник).

c4_12.

Дано предложение, которое состоит из слов, написанных строчными и прописными латинскими буквами. Слова разделены пробелами. В конце предложения стоит точка. Узнать, можно ли из всех символов данного предложения, исключая пробелы и точку, составить слово-перевертыш.

Например, из предложения Do not to do. можно составить слово-перевертыш Dootntood

Если это возможно, то напечатать на экран YES и один из примеров слова-перевертыша, иначе – NO.

При чтении примера слова-перевертыша слева-направо и справа-налево регистр буквы не учитывать, но при печати ответа на экран регистр буквы должен быть учтен.

с4_13.

В командных олимпиадах по программированию для решения предлагается не больше 12 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы на проверку, чтобы определить популярность той или иной задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием сети Интернет.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших запросов N. В каждой из последующих строк записан номер задачи от 1 до 12.

Пример входных данных:

```
6
1
2
1
1
5
2
```

Программа должна вывести список всех задач, встречающихся в запросах, в порядке возрастания (неубывания) количества запросов на проверку той или иной задачи с указанием количества запросов по ней. При этом каждая задача должна быть выведена ровно один раз вне зависимости от того, сколько раз встречается она в списке.

Пример выходных данных для выше приведенных входных данных:

```
5 1
2 2
1 3
```

с4_14.

Пользователю хостинговой компании доступны не более 13 доменов, у каждого домена – уникальное имя. Сервер хостинговой компании регистрирует обращение к каждому домену, сохраняя имя домена и количество переданных ему пакетов. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет составлять отчет по пришедшим на сервер запросам и определять количество доменов на которые было переслано минимальное количество пакетов за истекший период. Следует учитывать, что количество запросов в списке может быть очень велико. Перед тестом программы кратко опишите используемый вам и алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших запросов N . В каждой из последующих N строк записаны имя домена и количество переданных ему пакетов, разделенные двоеточием.

Длина строки не превосходит 100 символов, имя домена может содержать только буквы, цифры и символ "." (точка).

Пример входных данных:

```
5
Mydomen.homedom.ru:480
123.11.12:1200
123.ru:560
Mydomen.homedom.ru:80
123.11.12:140
```

Программа должна вывести имена всех доменов, встречающихся в списке, в порядке возрастания (не убывания) числа переданных на них пакетов, с указанием общего количества полученных доменом пакетов и количества доменов с минимальным количеством полученных пакетов. При этом каждый домен должен быть выведен только один раз.

Пример выходных данных для приведенного выше примера входных данных:

```
Mydomen.homedom.ru 560
123.ru 560
123.11.12 1340
2
```

с4_15.

Пользователю оператора голосовой Интернет-связи доступны не более 10 номеров, у каждого номера – уникальный логин. Система полностью доверяет своим клиентам, поэтому баланс каждого номера (целое число) может быть отрицательным. Каждому но-

меру соответствует отдельный счет, хранящий информацию о балансе на нем. При использовании удаленного доступа к системе сервер регистрирует логин пользователя и величину изменения счета. Предполагается, что начальный баланс каждого номера равен нулю.

Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет составлять отчет по хранимым пользовательским номерам, баланс которых в результате всех операций стал отрицательным.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших вопросов на изменение счета N . В каждой из последующих N строк записаны уникальный пользовательский логин и целое число – величина изменения, разделенные пробелом. Длина строки не превосходит 100 символов, логин пользователя может содержать буквы и цифры.

Пример входных данных:

```
5
234001 100
123000 -10
234001 -156
DD101101 123
123000 500
```

Программа должна вывести уникальные имена всех счетов, встречающихся в списке, в порядке убывания (не возрастания) количества средств на счете. При этом каждый счет должен быть выведен ровно один раз. После вывода баланса пользовательских счетов программа должна вывести количество счетов с отрицательным балансом.

Пример выходных данных для приведенного выше примера входных данных:

```
123000 490
DD101101 123
234001 -56
1
```

c4_15v2. Пользователю оператора голосовой Интернет-связи доступны не более 10 номеров, у каждого номера – уникальный логин. Система полностью доверяет своим клиентам, поэтому баланс каждого номера (целое число) может быть отрицательным. Каждому но-

меру соответствует отдельный счет, хранящий информацию о балансе на нем. При использовании удаленного доступа к системе сервер регистрирует логин пользователя и величину изменения счета. Предполагается, что начальный баланс каждого номера равен нулю.

Полный текст задачи смотрите выше.

с4_16.

В командных олимпиадах по программированию для решения предлагается не больше 11 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы, чтобы определить популярность той или иной задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается количество пришедших запросов N . В каждой из последующих N строк записано название задачи в виде текстовой строки. Длина строки не превосходит 100 символов, название может содержать буквы, цифры, пробелы и знаки препинания.

Пример входных данных:

```
6
A + B
Крестики-Нолики
Прямоугольник
Простой делитель
A + B
Простой делитель
```

Программа должна вывести список из трех наиболее популярных задач с указанием количества запросов по ним. Если в запросах упоминаются менее трех задач, то выведите информацию об имеющихся задачах. Если несколько задач имеют ту же частоту встречаемости, что и третья по частоте встречаемости задача, их тоже нужно вывести.

Пример выходных данных
для выше приведенных входных данных:

А + В 2
Простой делитель 2
Крестики-Нолики 1
Прямоугольник 1

с4_16v2. В командных олимпиадах по программированию для решения предлагается не больше 11 задач. Команда может решать предложенные задачи в любом порядке. Подготовленные решения команда посылает в единую проверяющую систему соревнований. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет статистически обрабатывать пришедшие запросы, чтобы определить наиболее популярные задачи. Следует учитывать, что количество запросов в списке может быть очень велико, так как многие соревнования проходят с использованием Интернет.

Полный текст условия задачи смотрите выше.

с4_17. Соревнования по игре "Тетрис-онлайн" проводятся по следующим правилам:

1. Каждый участник регистрируется на сайте игры под определенным игровым именем. Имена участников не повторяются.
2. Чемпионат проводится в течение определенного времени. В любой момент этого времени любой зарегистрированный участник может зайти на сайт чемпионата и начать зачетную игру. По окончании игры ее результат (количество набранных очков) фиксируется и заносится в протокол.
3. Участники имеют право играть несколько раз. Количество попыток одного участника не ограничивается.
4. Окончательный результат участника определяется по одной, лучшей для данного участника игре.
5. Более высокое место в соревнованиях занимает участник, показавший лучший результат.
6. При равенстве результатов более высокое место занимает участник, раньше показавший лучший результат.

В ходе соревнований заполняется протокол, каждая строка которого описывает одну игру и содержит результат участника и его игровое имя. Протокол формируется в реальном времени по ходу проведения чемпионата, поэтому строки в нем расположены в порядке проведения игр: чем раньше встречается строка в протоколе, тем раньше закончилась соответствующая этой строке игра.

Спонсор чемпионата предоставил призы различной ценности для награждения K лучших игроков ($K \leq 20$). Если участников окажется меньше K , призами награждаются все. Вам необходимо написать эффективную, в том числе по памяти, программу, которая по данным протокола определяет K лучших игроков и занятые ими места.

Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

Первая строка содержит числа: K – количество имеющихся призов и N – общее количество строк протокола. Каждая из следующих N строк содержит записанные через пробел результат участника (целое положительное число, не превышающее 100 миллионов) и игровое имя (имя не может содержать пробелы). Строки исходных данных соответствуют строкам протокола и расположены в том же порядке, что и в протоколе.

Описание выходных данных

Программа должны вывести имена и результаты K лучших игроков в порядке занятых мест по форме, приведенной ниже в примере. Если игроков окажется меньше K , нужно вывести данные обо всех игроках.

Пример входных данных:

```
6 15
69485 Jack
95715 qwerty
95715 Alex
83647 M
197128 qwerty
95715 Jack
93289 Alex
95715 Alex
95715 M
32768 BilboBaggins
99824 TetrisMaster
45482 BilboBaggins
62123 BilboBaggins
77623 M
```

56791 Champion

Пример выходных данных

для приведенного выше примера входных данных:

1. qwerty (197128)
2. TetrisMaster (99824)
3. Alex (95715)
4. Jack (95715)
5. M (95715)
6. BilboBaggins (62123)

- c4_17v2.** Соревнования по игре "Тетрис-онлайн" проводятся по следующим правилам:

Полный текст условия задачи смотрите выше.

- c4_17v3.** Соревнования по игре "Тетрис-онлайн" проводятся по следующим правилам:

Полный текст условия задачи смотрите выше.

- c4_18.** В офисном здании для обеспечения внутренней телефонной связью компаний арендаторов продумана некоторая схема учета абонентов АТС. Структура файла содержащего эту информацию следующая. В строке последовательно для каждого внутреннего номера записаны: - фамилия, имя и отчество абонента (каждое из слов не более 15 символов),
- наименование арендатора (до 15 символов) и его пятизначный номер договора, состоящий из цифр и букв,
- внутренний цифровой трехзначный номер.
Все записи в строке разделены ровно одним пробелом.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить фамилии абонентов и через пробел номера телефонов в порядке увеличения внутреннего номера по запросу от пользователя наименования арендатора. Известно, что все внутренние номера начинаются с цифры "9".

Пример строки файла:

Шульц Виктор Григорьевич СтройКарта А712В 912

Пример входной строки:

СтройКарта

Пример выходных данных:

Андреев 904

Яковлев 910

Шульц 912

с4_19.

Дан список точек плоскости с целочисленными координатами. Назовем удаленностью точки расстояние от этой точки до более далекой от нее оси координат. Необходимо определить:

- 1) номер координатной четверти K , в которой находится наименьшее ненулевое количество точек;
- 2) точку A в этой четверти, удаленность которой максимальна;
- 3) удаленность этой точки R .

Если в нескольких четвертях расположено одинаковое количество точек, следует выбрать ту четверть, в которой величина R больше.

При равенстве и количества точек, и величины R необходимо выбрать четверть с меньшим номером K .

Если в выбранной четверти несколько точек имеют одинаковую максимальную удаленность, нужно выбрать первую по списку.

Точки, хотя бы одна из координат которых равна нулю, считаются не принадлежащими ни одной четверти и не рассматриваются.

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N . Каждая из следующих N строк содержит координаты очередной точки – два целых числа (первое – координата x , второе – координата y).

Описание выходных данных

Программа должна вывести номер выбранной четверти K , количество точек в ней M , координаты выбранной точки A и ее удаленность R по образцу, приведенному ниже в примере.

Пример входных данных:

7

-3 6

1 2
1 1
3 0
4 -3
-6 8
-12 5

Пример выходных данных

для приведенного выше примера входных данных

$K = 4$
 $M = 1$
 $A = (4, -3)$
 $R = 4$

c4_20.

От цифровых датчиков в компьютер поступает информация о характеристиках физического процесса. Результатом каждого измерения является целое число.

Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет выводить среднее арифметическое поступивших измерений, без учета минимальных значений измерений.

Среднее арифметическое вычисляется и выводится как вещественное число в любом формате.

Следует учитывать, что количество измерений может быть очень велико. Также известно, что суммирование всех измерений не приведет к переполнению переменной целочисленного типа.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подается общее количество N значений измерений. В каждой из последующих N строк записано целое число. Гарантируется, что во входных данных есть как минимум два различных значения измерений.

Пример входных данных:

5
100
10
100
10
100

Пример выходных данных для приведенного выше примера входных данных:

100.0

3 Основные алгоритмы [alg_40]

- alg_1.** Даны действительные числа x , y . Получить минимум и максимум двух данных чисел.
- alg_2.** Найти минимум из трех данных чисел.
- alg_3.** Найти минимум из четырех данных чисел.
- alg_4.** Демонстрация работы с записями. (1 урок на "записи")
- alg_5.** Дано натуральное число, записанное в 10-ой системе счисления. Записать это же число в системе счисления с основанием, меньшим 10.
- alg_6.** Найти наименьший простой делитель данного натурального числа.
- alg_7.** Заполните одномерный массив по правилам:
а) 1, 3, 1, 3, ..., 1, 3;
б) 1, 3, 5, 7, ..., 131;
в) в элементы массива с нечетными индексами записать квадрат их индекса, а в остальные записать 0.
- alg_8.** Сформировать одномерный массив из первых N чисел Фибоначчи. Напомним, что числа Фибоначчи вычисляются по следующей формуле:
- $$F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}.$$
- alg_9.** Заполнить двумерный массив по следующему правилу: элементы главной диагонали равны 1, ниже главной диагонали – 0, а выше – сумме индексов.
- alg_10.** Дан двумерный массив. Все элементы, расположенные под побочной диагональю поменять на соответствующие им элементы, расположенные над побочной диагональю.
- alg_11.** Отсортировать массив методом "пузырька" (обменом). Пузырьковая сортировка.
- alg_11v2.** Произвести сортировку одномерного массива методом "пузырь-

ка".

- alg_12. Отсортировать массив методом выбора.
- alg_13. Отсортировать массив методом вставки (включения).
- alg_14. Методом вставок отсортировать целочисленный массив в возрастающем порядке.
- alg_15. Разложить натуральное число на простые множители.
- alg_16. Узнать, сколько слов входит в данное предложение. Каждое слово предложения напечатать в отдельной строке. Разделителем слов является пробел.
- alg_17. Написать программу, которая во введенной с клавиатуры строке (содержащей только буквы и пробелы) считает количество слов, начинающихся и заканчивающихся одной и той же буквой. Слово, состоящее из одной буквы, также считается. Между словами один пробел.
- alg_17v2. Дана строка, состоящая только из букв и пробелов. Посчитать, сколько в ней таких слов, у которых первая и последняя буква слова совпадают. Слово, состоящее из одной буквы считается также.
- alg_18. Дана строка символов. Переставить местами слова с максимальной и минимальной длиной.
- alg_19. Из заданного предложения выбрать и напечатать только те символы, которые встречаются в нем только раз (в том порядке, в котором они встречаются в тексте).
- alg_20. Написать программу, которая по введенному числу N ($2 < N < 100$) выводит без повторов все трехэлементные подмножества множества чисел $[1, 2, \dots, N]$.
- alg_21. Написать алгоритм линейного поиска элемента в массиве.
- alg_22. Написать программу линейного поиска элемента одномерного массива с использованием "барьера".
- alg_23. Бинарный поиск.

Дан массив A , элементы которого отсортированы в возрастающем порядке. Методом половинного деления найти данное число X в массиве A .

alg_23v2. Бинарный поиск числа. Написать программу, используя процедуры и функции.

alg_24. Дан массив целых чисел, заполненный случайными числами. Удалить все элементы, начиная с k_1 -го по k_2 -ой (k_1 и k_2 вводятся с клавиатуры). Сделать проверку корректности ввода значений k_1 и k_2 ($k_1 \leq k_2$), если ввод некорректный, то вывести сообщение об ошибке и закончить работу.

alg_25. Дан целочисленный массив A длиной N , в котором значениями заполнены первые K элементов ($K < N$). Дан также целочисленный массив B длиной R . Требуется вместо каждого равного 0 элемента из A среди K первых элементов вставить всю последовательность B .

alg_26. Дан одномерный массив из целых чисел. Перевернуть его.

alg_27. Заполнить массив при помощи генератора случайных чисел так, чтобы значения элементов массива принадлежали интервалу $[-10..34]$. Найти сумму квадратов всех элементов полученного массива.

alg_28. Дан целочисленный массив A . Найти максимальное значение в данном массиве и количество элементов, равных ему, за однократный просмотр массива.

alg_29. Даны действительные числа a, b, c . Вычислите корни квадратного уравнения

$$ax^2 + bx + c = 0.$$

alg_30. Даны действительные числа a, b, c . Вычислите корни уравнения

$$ax^2 + bx + c = 0.$$

alg_31. Написать программу слияния двух упорядоченных массивов.

alg_31v2. Сортировка массива методом "слияния".

- alg_32.** Дана последовательность целых чисел, длина которой не менее двух. Найти произведение двух максимальных чисел этой последовательности.
- alg_33.** Найти сумму двух минимальных элементов целочисленной последовательности. В последовательности не менее 2-х элементов.
- alg_34.** В данном массиве найти второе по величине значение (второе максимальное или второе минимальное) за однократный просмотр массива.
- alg_35.** Найти наибольший общий делитель двух натуральных чисел. Алгоритм Евклида.
- alg_36.** Найти наименьшее общее кратное двух натуральных чисел. Обозначим через NOK - наименьшее общее кратное, а через NOD - наибольший общий делитель. Тогда
- $$NOK(a, b) = \frac{a \cdot b}{NOD(a, b)}.$$
- alg_37.** Даны натуральные числа k, a_1, \dots, a_k , где $k \geq 2$. Вычислить наибольший общий делитель $NOD(a_1, \dots, a_k)$, воспользовавшись алгоритмом Евклида.
- alg_38.** Даны натуральные числа k, a_1, \dots, a_k , где $k \geq 2$. Вычислить наименьшее общее кратное $NOK(a_1, \dots, a_k)$, воспользовавшись алгоритмом Евклида.
- alg_39.** Вычислить произведение 2-х многочленов.
- alg_40.** Сколько существует четырехзначных чисел, в записи которых встречается не более двух различных цифр.
- alg_40v2.** Сколько существует четырехзначных чисел, в записи которых использовано не более двух различных цифр.

4 Задача 25, ЕГЭ-2016 [z25_2]

- z25_1.** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Напишите алгоритм, который находит и выводит сумму элементов наибольшей возрастающей последовательности подряд идущих элементов массива.
- z25_2.** Дан целочисленный массив, в котором не менее 2-х элементов. Элементы массива могут принимать произвольное значение. Напишите алгоритм, который находит и выводит значение второго максимума (элемента, который в отсортированном по невозрастанию массиве стоял бы вторым).