

Типы данных

1) Целые числа

Задача №1433. Кролики

Всем известен, так называемый, принцип Дирихле, который формулируется следующим образом:

Предположим, что некоторое число кроликов рассажены в клетках. Если число кроликов больше, чем число клеток, то хотя бы в одной из клеток будет больше одного кролика.

В данной задаче мы рассмотрим более общий случай этого классического математического факта. Пусть есть n клеток и m зайцев, которых рассадили по этим клеткам. Вам требуется рассчитать максимальное количество зайцев, которое гарантированно окажется в одной клетке.

Входные данные

В первой строке входного файла записаны два натуральных числа n и m . ($1 \leq n, m \leq 10^9$).

Выходные данные

В выходной файл выведите ответ на задачу.

Примеры

входные данные

```
2 3
```

выходные данные

```
2
```

Задача №1466. Сумма от 1 до N

Сумму всех целых чисел от 1 до 100 можно посчитать при помощи хитрого приема. Разобьем все числа по парам 1 и 100, 2 и 99, 3 и 98 и т.д. Сумма каждой пары 101. Пар всего 100 пополам (50). Поэтому сумма равна $\frac{(1+100) \cdot 100}{2}$.

Для нечетного количества слагаемых работает та же формула: например, $1+2+3 = \frac{(1+3) \cdot 3}{2} = 6$.

Входные данные

Одно целое число N . Гарантируется, что ответ "помещается" в тип int64 (Паскаль) / long long (в Си) / long (в Java).

Выходные данные

Одно число – сумма всех целых чисел от 1 до N .

Примеры

входные данные

100

выходные данные

5050

входные данные

3

выходные данные

6

Задача №1475. k -я секунда суток

Идёт k -я секунда суток. Определите, сколько целых часов h и целых минут m прошло с начала суток.

Например, если

$$k=13257=3\cdot3600+40\cdot60+57,$$

то $h=3$ и $m=40$.

Входные данные

На вход программе подается целое число k ($0 \leq k \leq 86399$).

Выходные данные

Выведите на экран фразу:

It is ... hours ... minutes.

Вместо многоточий программа должна выводить значения h и m , отделяя их от слов ровно одним пробелом.

Примеры

входные данные

13257

выходные данные

It is 3 hours 40 minutes.

Задача №1476. Часовая стрелка

Часовая стрелка повернулась с начала суток на d градусов. Определите, сколько сейчас целых часов h и целых минут m .

Входные данные

На вход программе подается целое число d ($0 \leq d < 360$).

Выходные данные

Выведите на экран фразу:

```
It is ... hours ... minutes.
```

Вместо многоточий программа должна выводить значения h и m , отделяя их от слов ровно одним пробелом.

Примеры

входные данные

```
90
```

выходные данные

```
It is 3 hours 0 minutes.
```

Задача №1479. Без циклов

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 $(2k)$ -ю и т. д. Напишите программу, по номеру строки в тексте определяющую номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

Входные данные

На вход программе подаются число k — количество строк на странице и число n — номер строки в тексте ($1 \leq k \leq 200$, $1 \leq n \leq 20000$).

Выходные данные

Выведите два числа — номер страницы, на которой будет напечатана эта строка, и номер строки на этой странице

Для считывания данных на языке Python 3 можно использовать следующую конструкцию:

```
k, n = map(int, input().split())
```

Примеры

входные данные

```
50 1
```

выходные данные

```
1 1
```

входные данные

20 25

выходные данные

2 5

входные данные

15 43

выходные данные

3 13

Задача №1483. Два момента времени

Даны значения двух моментов времени, принадлежащих одним и тем же суткам: часы, потом минуты и секунды для каждого из моментов времени. Известно, что второй момент времени наступил не раньше первого. Определите, сколько секунд прошло между двумя моментами времени.

Входные данные

В первой строке входных данных находятся три целых числа — часы, минуты и секунды первого момента времени. Во второй строке — три числа, характеризующие второй момент времени. Число часов лежит в диапазоне от 0 до 23, число минут и секунд — от 0 до 59.

Выходные данные

Выведите число секунд между двумя моментами времени.

Примеры**входные данные**

1 1 1

2 2 2

выходные данные

3661

входные данные

1 2 30

1 3 20

выходные данные

50

2) Символы и строки

Символы и строки

До сих пор мы работали только с числовыми данными, как целыми, так и действительными. Но, как правило (если речь идет не о сугубо научных расчетах) приходится оперировать с текстовой информацией. В этом листочке содержится введение в методы обработки текстовых данных.

Символьный тип `char`

Любой текст состоит из символов. Для хранения одного символа предназначен тип данных `char`. Переменную типа `char` можно рассматривать двояко: как целое число, занимающее 1 байт и способное принимать значения от 0 до 255 (тип `unsigned char`) или от -128 до 127 (тип `signed char`) и как один текстовый символ. Сам же тип `char` может оказаться как знаковым, так и беззнаковым, в зависимости от операционной системы и компилятора. Поэтому использовать тип `char` не рекомендуется, лучше явно указывать будет ли он знаковым (`signed`) или беззнаковым (`unsigned`).

Как и целые числа, данные типа `char` можно складывать, вычитать, умножать, делить, а можно выводить на экран в виде одного символа. Именно это и происходит при выводе символа через объект `cout`. Если же нужно вывести числовое значение символа (также называемое ASCII-кодом), то значение символа необходимо преобразовать к типу `int`. Например:

```
#include<iostream>
using namespace std;
int main()
{
    unsigned char c='A'; // Константы char заключаются в одинарные кавычки
    cout<<c<<" "<<(int)c<<endl;
    c=126; // char можно присвоить и числовое значение
    cout<<c<<" "<<(int)c<<endl;
    return 0;
}
```

В этом примере переменной `c` типа `char` присваивается значение, равное символу 'A' (константы типа `char` записываются как символы в одинарных кавычках), затем на экран выводится значение `c`, как символа и его ASCII-код, потом переменной `c` присваивается значение 126 (то есть символ с ASCII-кодом 126) и снова выводится на экран символ и его ASCII-код.

Организовать последовательное посимвольное считывание всего входного потока можно при помощи цикла `while`:

```
#include<iostream>
using namespace std;
int main()
{
    unsigned char c;
    while(cin>>c) // Цикл пока считывание успешно
    { // Делаем необходимые действия
    }
```

```

    return 0;
}

```

В этом примере программа будет посимвольно считывать входной поток (по умолчанию — ввод с клавиатуры), пока не встретит признак конца файла. Для того, чтобы сообщить программе о завершении файла при вводе с клавиатуры необходимо нажать клавиши `Ctrl-d` в системе Linux и `Ctrl-z` в системе Windows.

Эта программа при считывании данных будет игнорировать символы-разделители: пробелы, символы новой строки и табуляции. Если нужно, чтобы в переменную `s` считывались все символы, в том числе и разделители, то необходимо для потока ввода `cin` установить манипулятор `noskipws` при помощи инструкции `cin>>noskipws;`.

Строки в языке C++

Текстовую строку можно представить, как массив символов типа `char`, но в языке C++ для хранения текстовых строк был создан более удобный тип `string`. По сути, тип данных `string` и является массивом символов. Например, если мы объявили переменную `S` как `string S`, а затем присвоили ей значение `"школа"` (текстовые строки заключаются в двойные кавычки), то мы можем обращаться к отдельным символам строки `S`, представляя `S`, как массив символов, например, `S[0]=='ш'`, `S[1]=='к'` и т.д. Для того, чтобы узнать длину строки используется метод `length()`, вызываемый в виде `S.length()`.

Строковые данные можно считывать с клавиатуры, выводить на экран, присваивать переменным типа `string`. Также строки можно складывать друг с другом: например, при сложении строк `"Hello, "` и `"world!"` получится строка `"Hello, world!"`. Такая операция над строками называется *конкатенацией*.

Основные приемы работы с объектами `string` проиллюстрированы в программе:

```

string S, S1, S2;           // Объявление трех строк
cout<<"Как вас зовут? ";
cin>>S1;                    // Считали строку S1
S2="Привет, ";              // Присвоили строке значение
S=S2+S1;                    // Использование конкатенации
cout<<S<<endl;              // Вывод строки на экран
cout<<S.length();           // Длина строки S

```

При считывании строк из входного потока считываются все символы, кроме символов-разделителей (пробелов, табуляций и новых строк), которые являются границами между строками. Например, если при выполнении следующей программы

```

string S1, S2, S3; // объявили 3 строки
cin>>S1>>S2>>S3;

```

ввести текст ``Мама мыла раму'` (с произвольным количеством пробелов между словами), то в массив `S1` будет записана строка `"Мама"`, в `S2` — `"мыла"`, в `S3` — `"раму"`.

Таким образом, организовать считывание всего файла по словам, можно следующим образом:

```

string s;
while(cin>>s) // Цикл пока считывание успешно

```

```
{
    // Делаем необходимые действия
}
```

Если нужно считать строку со всеми пробелами, то необходимо использовать функцию `getline` следующим образом:

```
string S;
getline(cin, S);
```

В данном случае если запустить эту программу и ввести строку "Мама мыла раму", то именно это значение и будет присвоено строке `S`. Считать же весь входной поток по строкам можно при помощи следующего кода:

```
string s;
while ( getline(cin, S) ) // Цикл пока считывание успешно
{
    // Делаем необходимые действия
}
```

Упражнения

Если в упражнении сказано "дана строка", то ее можно считать при помощи инструкции вида `cin>>S` или при помощи функции `getline`. Если сказано, что дана строка, содержащая пробелы, то ее необходимо считывать при помощи функции `getline`.

1. (Без тестирующей системы) Выведите на экран все возможные символы с ASCII-кодами от 32 до 255 (символы с кодами до 32 являются служебными и их вывод на экран может привести к забавным последствиям) и их ASCII-коды. Формат вывода: сначала ASCII-код, затем символ, например,

```
32
33 !
34 "
35 #
```

2. (A) Напишите функцию `bool IsDigit(unsigned char c)`, определяющую, является ли данный символ цифрой или нет. Напишите программу, которая получает на вход один символ и выводит строку `yes`, если символ является цифрой и строку `no`, в противном случае.
3. (B) Программа получает на вход один символ. Если этот символ является строчной буквой латинского алфавита (то есть буквой от `a` до `z`), выведите вместо него аналогичную заглавную букву, иначе выведите тот же самый символ (такая операция называется "перевод в верхний регистр"). Для этого сделайте функцию `unsigned char ToUpper(unsigned char c)`, которая переводит данный символ в верхний регистр.
4. (C) Измените регистр символа: если он был латинской буквой: сделайте его заглавным, если он был строчной буквой и наоборот. Для этого напишите отдельную функцию, меняющую регистр символа.
5. (D) Даны две строки. Определите, совпадают ли они сравнив их посимвольно. Напишите для этого функцию `bool Compare(string S1, string S2)`.

Вход: две строки. **Выход:** слово `yes`, если строки совпадают, слово `no` в противном случае.

6. (E) Дана строка, содержащая пробелы. Найдите, сколько в нем слов (слово – это последовательность непробельных символов, слова разделены одним пробелом, первый и последний символ строки – не пробел).

Вход: На вход подается несколько строк. Входная строка должна считываться методом `getline`. **Выход:** количество слов в первой из введенных строк.

7. (F) Дана строка, содержащая пробелы. Найдите в ней самое длинное слово, выведите на экран это слово и его длину.

Вход: одна строка, содержащая пробелы. Слова разделены ровно одним пробелом. Строка должна считываться методом `getline` (программа должна считывать только одну первую строку). **Выход:** самое длинное слово в строке и его длина.

8. (G) По данной строке, определите, является ли она палиндромом (то есть можно ли прочесть ее наоборот, как, например, слово ``топот'`).

Вход: одна строка без пробелов. **Выход:** `yes`, если слово является палиндромом, `no` в противном случае.

9. (H) Дана строка. Известно, что она содержит ровно две одинаковые буквы. Найдите эти буквы.

Вход: одна строка. **Выход:** одна буква, которая встречается в строке дважды.

10. (I) Даны две строки. Определите, является ли первая строка подстрокой второй строки.

Вход: две строки. **Выход:** слово `yes`, если первая строка является подстрокой второй строки, или слово `no` в противном случае.

11. (J) Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: `"North 5"`, где первое слово – одно из `"North"`, `"South"`, `"East"`, `"West"`, а второе число – количество шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось `OX` направлена на восток, ось `OY` – на север.

Вход: последовательность строк указанного формата. **Выход:** координаты клада – два целых числа через пробел.

Например, при вводе

```
North 5
East 3
South 1
```

программа должна вывести координаты **3 4**.

12. (K) Дана строка, содержащая пробелы. Проверьте, является ли она палиндромом без учета пробелов (например, ``аргентина манит негра'`).

Вход: одна строка, содержащая пробелы. Подряд может идти произвольное число пробелов. **Выход:** `yes`, если данная строка является палиндромом и `no` в противном случае.

Задача №102. Проверить, является ли символ цифрой

Напишите функцию `boolean IsDigit(char c)` (Java), `function IsDigit(c:char):boolean` (Pascal), `bool IsDigit(unsigned char c)` (C/C++), определяющую, является ли данный символ цифрой или нет.

Естественно, программа должна считывать данные, вызывать эту функцию и выдавать ответ.

Входные данные

Задан единственный символ `c`.

Выходные данные

Необходимо вывести строку `yes`, если символ является цифрой, и строку `no` в противном случае.

Примеры

входные данные

`c`

выходные данные

`no`

входные данные

`2`

выходные данные

`yes`

Задача №103. Перевести символ в верхний регистр

Напишите функцию `unsigned char ToUpper(unsigned char c)` (C/C++), `function ToUpper(c:char):char` (Pascal), которая переводит данный символ в верхний регистр.

Входные данные

Задан один символ `c`.

Выходные данные

Если символ является строчной буквой латинского алфавита (то есть буквой от `a` до `z`), выведите вместо него аналогичную заглавную букву, иначе выведите тот же самый символ (такая операция называется "перевод в верхний регистр").

Примеры

входные данные

`e`

выходные данные

E

входные данные

B

выходные данные

B

Задача №104. Изменить регистр символа

Измените регистр символа, если он был латинской буквой: сделайте его заглавным, если он был строчной буквой и наоборот. Для этого напишите отдельную функцию, меняющую регистр символа.

Входные данные

Задан единственный символ **C**.

Выходные данные

Необходимо вывести получившийся символ.

Примеры

входные данные

q

выходные данные

Q

входные данные

W

выходные данные

w

Задача №105. Совпадают ли строки?

Напишите функцию `bool Compare(string S1, string S2)`, определяющую, совпадают ли 2 строки.

Входные данные

Заданы 2 строки.

Выходные данные

Необходимо вывести слово `yes`, если строки совпадают, и слово `no` в противном случае.

Примеры

входные данные

```
a
```

```
a
```

ВЫХОДНЫЕ ДАННЫЕ

```
yes
```

Задача №106. Количество слов

Дана строка, содержащая пробелы. Найдите, сколько в ней слов (слово – это последовательность непробельных символов, слова разделены одним пробелом, первый и последний символ строки – не пробел).

Входные данные

На вход подается несколько строк.

Выходные данные

Необходимо вывести количество слов в первой из введенных строк.

Примеры

входные данные

```
In the town where I was born
```

выходные данные

```
7
```

Задача №107. Самое длинное слово

Дана строка, содержащая пробелы. Найдите в ней самое длинное слово, выведите это слово и его длину. Если таких слов несколько, выведите первое из них.

Входные данные

Задана одна строка, содержащая пробелы. Слова разделены ровно одним пробелом. Пробелы в начале и конце строки *допускаются*.

Выходные данные

Необходимо вывести самое длинное слово в строке и его длину.

Примеры

входные данные

```
one two three four five six
```

выходные данные

```
three
```

```
5
```

Задача №108. Является ли слово палиндромом?

По данной строке определите, является ли она палиндромом (то есть, можно ли прочесть ее наоборот, как, например, слово "топот").

Входные данные

На вход подается 1 строка без пробелов.

Выходные данные

Необходимо вывести `yes`, если строка является палиндромом, и `no` в противном случае.

Примеры

входные данные

```
abba
```

выходные данные

```
yes
```

Задача №109. Две одинаковые буквы

Дана строка. Известно, что она содержит ровно две одинаковые буквы. Найдите эти буквы. Гарантируется, что повторяются буквы только одного вида.

Входные данные

На вход подается 1 строка.

Выходные данные

Необходимо вывести букву, которая встречается в строке дважды.

Примеры

входные данные

```
fif
```

выходные данные

```
f
```

Задача №110. Поиск подстроки

Даны две строки. Определите, является ли первая строка подстрокой второй строки.

Входные данные

На вход подается 2 строки длины не более 10000, состоящие только из маленьких букв латинского алфавита.

Выходные данные

Необходимо вывести слово `yes`, если первая строка является подстрокой второй строки, или слово `no` в противном случае.

Примеры

входные данные

```
abac  
ababacaba
```

выходные данные

```
yes
```

Задача №111. Капитан Флинт

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: "North 5", где слово – одно из "North", "South", "East", "West", – задает направление движения, а число – количество шагов, которое необходимо пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось OX направлена на восток, ось OY – на север.

Входные данные

На вход подается последовательность строк указанного формата. Гарантируется, что числа не превосходят 10⁸.

Выходные данные

Необходимо вывести координаты клада – два целых числа через пробел. Гарантируется, что эти числа не превосходят 10⁸.

Примеры

входные данные

```
South 19
```

выходные данные

```
0 -19
```

Задача №112. Является ли строка палиндромом?

Дана строка, состоящая из строчных латинских букв и пробелов. Проверьте, является ли она палиндромом без учета пробелов (например, "аргентина манит негра").

Входные данные

На вход подается 1 строка длины не более 100, содержащая пробелы. Подряд может идти произвольное число пробелов.

Выходные данные

Необходимо вывести `yes`, если данная строка является палиндромом, и `no` в противном случае.

Примеры

входные данные

```
ab a
```

выходные данные

```
yes
```

Задача №1415. Шифр Юлия

Юлий Цезарь использовал свой способ шифрования текста. Каждая буква заменялась на следующую по алфавиту через K позиций по кругу. Необходимо по заданной шифровке определить исходный текст.

Входные данные

В первой строке дана шифровка, состоящая из заглавных латинских букв. Во второй строке число K ($1 \leq K \leq 10$).

Выходные данные

Требуется вывести результат расшифровки.

Примеры

входные данные

```
XPSE
```

```
1
```

выходные данные

```
WORD
```

входные данные

```
ZABC
```

```
3
```

выходные данные

```
WXYZ
```

Задача №1417. Конвертирование

Дана строка S , в которой выделили подстроку, состоящую из символов с i -го по j -й включительно (символы строки S нумеруются с единицы) и поменяли местами i -й символ с j -м, $(i+1)$ -й с $(j-1)$ -м и так далее (конвертировали подстроку). Выведите строку S после внесенных изменений.

Входные данные

В первой строке входного файла содержится строка S , длиной не более 1000 символов, во второй – числа i и j ($i \leq j$).

Выходные данные

В выходной файл выведите ответ на задачу.

Примеры

входные данные

```
vjhoamkts
7 8
```

выходные данные

```
vjhoamkts
```

Задача №1421. Удали пробелы

Дана строка, Вам требуется преобразовать все идущие подряд пробелы в один.

Входные данные

Длина строки не превосходит 1000.

Выходные данные

Выведите измененную строку.

Примеры

входные данные

```
nz d urp lren s bwz boom t a j ho vi
```

выходные данные

```
nz d urp lren s bwz boom t a j ho vi
```

входные данные

```
d iz czl l l h udq t
```

выходные данные

```
d iz czl l l h udq t
```

Задача №1435. IP-адрес

Для того чтобы выходить в Интернет, каждому компьютеру присваивается так называемый IP-адрес. Он состоит из четырех целых чисел в диапазоне от 0 до 255, разделенных точками. В следующих трех строках показаны три правильных IP-адреса:

127.0.0.0

192.168.0.01

255.00.255.255

Напишите программу, которая определяет, является ли заданная строка правильным IP-адресом.

Входные данные

На вход программе подается строка длиной не более 15 символов, которая включает цифры и ровно три точки.

Выходные данные

Если строка является правильным IP-адресом, необходимо вывести 1, иначе 0.

Примеры

входные данные

127.0.0.1

выходные данные

1

входные данные

12...34

выходные данные

0

Задача №1450. Идеальный отряд

Как вы помните, месклениты отправились в экспедицию. Однажды руководителю экспедиции потребовалось отправить на разведку специальный отряд, состоящих из лучших мескленитов. Для этого он выстроил всю команду в шеренгу.

Цвет панциря каждого мескленита обозначается заглавной латинской буквой (от "A" до "Z"). В целях экономии времени руководитель собирается выбрать из шеренги несколько подряд стоящих. Кроме того, он считает, что разведка будет более удачной, если выбранный отряд будет симметричен по цветам панцирей. Например, отряд "RGBGR" будет симметричным, а отряд "RGRB" – нет.

Требуется выбрать из шеренги мескленитов максимально возможный отряд, удовлетворяющий данным условиям.

Входные данные

Дана строка, длина которой не превосходит 255 символов – цвета мескленитов в шеренге.

Выходные данные

Выходные данные представляют собой строку – выбранный отряд мескленитов. Если возможных вариантов ответа несколько, то требуется вывести находящийся ближе к началу шеренги.

Примеры

входные данные

АВАС

выходные данные

АВА

входные данные

АВВСВ

выходные данные

ВСВ

Задача №2796. Перwokлассная задача

Преподаватель по программированию некоего Центра для одаренных детей, узнав, что его ученики знают математику 3-го класса на 97.001 процентов, решил проверить их знания по курсу математики 1-го класса. Для этого он взял за основу популярнейшую у математиков 1-го класса задачу. Перwokлассник должен был продолжить следующую последовательность рядов:

1

11

21

1211

111221

312211

13112221

Входные данные

В единственной строке входного файла записаны два целых числа через пробел: $x(0 \leq x \leq 100)$ - первый член последовательности и $n(1 \leq n \leq 25)$.

Выходные данные

Выведите n -ый ряд x -ой последовательности

Примеры

входные данные

1 4

выходные данные

1211

Задача №2797. Второклассная задача

У второклассников очень популярна следующая задача:

$$101=1$$

$$8181515=4$$

$$1111112=0$$

$$888888=14$$

$$1010101=3$$

$$7000007=?$$

Преподавателю программирования в некоем Центре для одаренных детей так понравилась эта задачка, что он решил именно ее использовать для проверки знаний математики 2-го класса у своих учеников.

Пусть первое число x , а соответствующее ему n

Входные данные

Единственное неотрицательное число x , не превышающее 10^{1001} .

Выходные данные

Выведите n .

Примеры

входные данные

689

выходные данные

4

Задача №2805. Интересный перевод

На планете Роботов очень не любят десятичную систему счисления, поэтому они попросили Вас написать программу, которая заменяет все встречающиеся в тексте числа на эти же числа, но в двоичной системе счисления.

Входные данные

Единственная строка, состоящая из любых символов. Длина строки не превышает 255 символов. Гарантируется, что во всех числах нет ведущих нулей.

Выходные данные

Выведите преобразованную строку.

Примеры

входные данные

```
6^&678JKjdkdl;?.,lk879Pk1kdf14839
```

выходные данные

```
110^&1010100110JKjdkdl;?.,lk1101101111Pk1kdf11001011100111
```

3) Структуры

Раньше для хранения данных использовались простые типы данных: числа, строки и т.д. Тем не менее, многие объекты, которые возникают в программировании, нельзя охарактеризовать только одной числовой или строковой величиной. Например, точка на плоскости задается парой действительных чисел (x, y), а данные о человеке можно задавать при помощи нескольких строк (фамилии, имени, отчества) и числового параметра: года рождения.

Поэтому удобным является использование объектов, объединяющих сразу ряд параметров, каждый из которых может фигурировать, как отдельная переменная.

Для этого используются специальные типы данных, которые в языке С называются *структурами*, а в языке С++ – *классами*. Например, структура `Point`, задающая точку на плоскости и содержащая два действительных числа `x` и `y` может быть задана следующим образом:

```
struct Point
{
    double x;
    double y;
};
```

Переменные `x` и `y`, входящие в структуру `Point`, называются *полями структуры*. Определение структуры дается вне всех функций (и, обычно делается перед объявлением всех функций). Определение структуры обязательно завершается точкой с запятой.

После этого мы можем работать с `Point`, как с новым типом данных, содержащим два поля: `x` и `y`. Примеры создания переменной и массива переменных типа `Point`:

```
Point P, Arr[10];
```

Чтобы обратиться к полю какой-либо структуры, используется оператор "точка". Его левый операнд – идентификатор переменной типа структура, правый операнд – имя поля. Например, `P.x` или `Arr[i].y`.

Например, считать координаты точки, сохранить их в переменной `P` и вывести на экран расстояние от начала координат до этой точки можно следующим образом:

```
cin>>P.x>>P.y;
cout<<sqrt(P.x*P.x+P.y*P.y);
```

По сути, величины `P.x` и `P.y` являются независимыми переменными, скомбинированными в один объект. С этим объектом можно работать, как с единым целым, например, можно выполнять присваивания вроде `Arr[i]=P`, можно сохранить набор точек в одном массиве и т.д.

Аналогично, можно определить структуру типа `Person` для хранения информации о человеке:

```
struct Person
{
    string FirstName; // Имя
    string LastName;  // Фамилия
    int     BirthYear; // Год рождения
};
```

Теперь можем работать с данными типа `Person`:

```
Person Vasya;
Vasya.FirstName="Василий";
Vasya.LastName="Пупкин";
Vasya.BirthYear=1990;
cout<<Vasya.FirstName<<" "<<Vasya.LastName<<
    " родился в "<<Vasya.BirthYear<<" году"<<endl;
```

Полями структур могут быть произвольные типы данных. Можно, например, создать структуру, полями которой будут массивы или другие структуры.

Например, пусть мы хотим определить структуру `Circle` для определения окружности. Окружность задается центром и радиусом. Радиус – это действительное число (поле `Radius` типа `double`), а центр – это, конечно же, точка, то есть поле `Center` имеет тип `Point`. Получили:

```
struct Circle
{
    Point Center;
    double Radius;
};
```

Дальше с такими "вложенными" структурами можно работать так:

```
Circle A;
A.Radius=10;
A.Center.x=-3;
A.Center.y=15;
```

Задачи

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Например, четыре точки в вершинах единичного квадрата задаются так:

```
4
0 0
0 1
1 1
1 0
```

Создайте структуру `Point` и сохраните исходные данные в массиве структур `Point`.

Если в задаче ответом является действительное число, его необходимо выводить с точностью в 15 значащих знаков (перед выводом необходимо дать инструкцию `cout.precision(15)`).

1. (A) Выведите координаты наиболее удаленной от начала координат точки.
2. (B) Выведите координаты центра тяжести данного множества точек (учтите, что это – два *действительных* числа).
3. (C) Выведите диаметр данного множества – максимальное расстояние между двумя данными точками. В реализации программы должна быть функция вычисления расстояния между двумя точками.
4. (D) Выведите все исходные точки в порядке возрастания их расстояний от начала координат. Программа выводит только последовательность точек, их количество выводить не надо.
5. (E) Среди исходных точек найдите три, образующие треугольник с максимальным периметром. Выведите данный периметр.
6. (F) Среди исходных точек найдите три, образующие треугольник максимальной площади. Выведите данную площадь.

Дан текстовый файл содержащий список учащихся и их оценки по трем предметам: математике, физике, информатике. Формат файла: сначала количество учащихся n , затем n строк, каждая из

которых содержит фамилию, имя и три числа. Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Пример входного файла:

```
4
Ivanov Vasiliy 5 3 4
Petrov Sergey 4 3 5
Konstantinov Nikolay 5 5 5
Kuznetsov Ivan 5 4 4
```

Если в программе требуется вывести фамилии и имена учащихся, то необходимо вывести их по одному на строке разделяя одним пробелом. Выводить оценки не нужно. Порядок вывода должен быть таким же, как в исходных данных, если не оговорено другое. Например:

```
Petrov Sergey
Kuznetsov Ivan
```

7. (G) Определите средний балл всех учащихся по каждому предмету. Выведите три действительных числа: средний балл всех учащихся по математике, по физике, по информатике.
8. (H) Выведите фамилии и имена учащихся, не имеющих троек (а также двоек и колов).
9. (I) Определите учащихся с наилучшей успеваемостью, то есть с максимальным средним баллом по трем предметам. Выведите одного или нескольких учащихся, имеющих максимальный средний балл.
10. (J) (Демо-версия ЕГЭ-2007 по информатике, задача C4) Определите трех учащихся с наилучшим средним баллом по трем предметам. Выведите фамилии и имена этих учащихся. Если при этом у нескольких учащихся средний балл совпадает со средним баллом учащегося "занявшего 3-е место", то необходимо вывести их всех (то есть если средние баллы учащихся равны 1, 2, 3, 4, 5, то нужно вывести трех учащихся со средним баллом 3, 4, 5, а если средние баллы равны 1, 2, 3, 4, 4, 5, 5, то нужно вывести четырех учащихся со средним баллом 4 или 5). Выводить фамилии и имена необходимо в порядке ввода.
11. (K) Выведите фамилии и имена учащихся в порядке убывания их среднего балла.

Задача №322. Наиболее удаленная точка

Выведите координаты наиболее удаленной от начала координат точки.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Выведите координаты точки, наиболее удаленной от начала координат.

Примеры

входные данные

2

```
1 2
2 3
```

выходные данные

```
2 3
```

Задача №323. Центр тяжести

Выведите координаты центра тяжести данного множества точек.

Создайте структуру `Point` и сохраните исходные данные в массиве структур `Point`.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Выведите координаты центра тяжести данного множества точек. Ответ необходимо выводить с точностью в 15 значащих цифр.

Примеры

входные данные

```
2
1 2
2 3
```

выходные данные

```
1.5 2.5
```

Задача №324. Диаметр множества

Выведите диаметр данного множества – максимальное расстояние между любыми двумя точками.

Создайте структуру `Point` и сохраните исходные данные в массиве структур `Point`.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа:

координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Необходимо вывести диаметр данного множества с точностью в 15 значащих цифр.

Примеры

входные данные

```
2
1 2
2 3
```

выходные данные

```
1.4142135623731
```

Задача №325. Сортировка точек

Выведите все исходные точки в порядке возрастания их расстояний от начала координат.

Создайте структуру `Point` и сохраните исходные данные в массиве структур `Point`.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Необходимо вывести все исходные точки в порядке возрастания их расстояний от начала координат. Программа выводит только координаты точек, их количество выводить не надо.

Примеры

входные данные

```
2
1 2
2 3
```

выходные данные

```
1 2
2 3
```

Задача №326. Максимальный периметр

Среди исходных точек найдите три, образующие треугольник с максимальным периметром. Выведите данный периметр.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n ($2 < n < 101$), затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Необходимо вывести найденный периметр с точностью в 15 значащих цифр.

Примеры

входные данные

```
4
0 0
0 1
1 0
1 1
```

выходные данные

```
3.41421356237309
```

Задача №327. Максимальная площадь

Среди исходных точек найдите три, образующие треугольник максимальной площади. Выведите данную площадь.

Создайте структуру Point и сохраните исходные данные в массиве структур Point.

Входные данные

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n ($2 < n < 101$), затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Все исходные координаты – целые числа, не превосходящие 10^3 .

Выходные данные

Необходимо вывести найденную площадь с точностью в 15 значащих цифр.

Примеры

входные данные

```
4
0 0
0 1
1 0
```

```
1 1
```

выходные данные

```
0.5
```

Задача №328. Средний балл по предметам

Определите средний балл всех учащихся по каждому предмету.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Выведите три действительных числа: средний балл всех учащихся по математике, по физике, по информатике.

Примеры

входные данные

```
2
Markov Valeriy 4 5 2
Kozlov Georgiy 5 1 2
```

выходные данные

```
4.5 3 2
```

Задача №329. Учащиеся без троек

Выведите фамилии и имена учащихся, не имеющих троек (а также двоек и колов).

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Порядок вывода должен быть таким же, как в исходных данных.

Примеры

входные данные

```
3
Babat Anna 5 4 3
```

```
Belova Galina 4 3 5
Moroz Yaroslav 3 5 4
```

выходные данные

Задача №331. Трое лучших

Определите трех учащихся с наилучшим средним баллом по трем предметам. Выведите фамилии и имена этих учащихся. Если при этом у нескольких учащихся средний балл совпадает со средним баллом учащегося, "занявшего 3-е место", то необходимо вывести их всех.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Порядок вывода должен быть таким же, как в исходных данных.

Примеры

входные данные

```
3
Yakovlev Ivan 5 5 5
Yapryntsev Aleksey 5 5 5
Kozlov Georgiy 5 5 5
```

выходные данные

```
Yakovlev Ivan
Yapryntsev Aleksey
Kozlov Georgiy
```

Задача №332. Отсортировать по среднему баллу

Выведите фамилии и имена учащихся в порядке убывания их среднего балла.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Если несколько учащихся имеют одинаковые средние баллы, то их нужно выводить в порядке, заданном во входных данных.

Примеры

входные данные

```
2
Markov Valeriy 1 1 1
Ivanov Ivan 2 2 2
```

выходные данные

```
Ivanov Ivan
Markov Valeriy
```

входные данные

```
3
Markov Valeriy 5 5 5
Sergey Petrov 1 1 1
Petrov Petr 3 3 3
```

выходные данные

```
Markov Valeriy
Petrov Petr
Sergey Petrov
```

Задача №1404. Личные дела

Однажды, неловкая секретарша перепутала личные дела учащихся. Теперь их снова необходимо упорядочить сначала по классам, а внутри класса по фамилиям

Входные данные

В первой строке дано число N ($1 \leq N \leq 1000$) – количество личных дел. Далее для каждого из N учащихся следующие данные (каждое в своей строке): фамилия и имя, класс, дата рождения. Фамилия и имя – строки не более чем из 20 символов, класс – строка состоящая из числа (от 1 до 11) и латинской буквы (от "A" до "Z"), дата рождения – дата в формате "ДД.ММ.ГГ" . Гарантируется, что внутри одного класса нет однофамильцев.

Выходные данные

В выходной файл требуется вывести N строк, в каждой из которых записаны данные по одному учащемуся. Строки должны быть упорядочены сначала по классам, а затем по фамилиям.

Примеры

входные данные

```
3
Sidorov
```

```
Sidor
9A
20.07.93
Petrov
Petr
9B
23.10.92
Ivanov
Ivan
9A
10.04.93
```

выходные данные

```
9A Ivanov Ivan 10.04.93
9A Sidorov Sidor 20.07.93
9B Petrov Petr 23.10.92
```

5) Действительные числа

Задача №74. $a + b = c$

Даны три действительных числа: a , b , c . Проверьте, выполняется ли равенство $a + b = c$. Если равенство выполняется, выведите YES, если не выполняется, выведите NO.

Входные данные

Числа a , b , c – действительные, положительные, не превосходят 10 и заданы не более, чем с 7 знаками после точки.

Выходные данные

Выведите результат сравнения.

Примеры

входные данные

```
2
3
7
```

выходные данные

```
NO
```

входные данные

0.1
0.2
0.3

выходные данные

YES

Задача №596. Утренняя пробежка - 1

В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал пробег на 70% от предыдущего значения. По данному числу y определите номер дня, на который пробег спортсмена составит не менее y километров.

Входные данные

На вход программа получает два действительных числа x и y . Числа положительные, действительные, не превосходят 1000, заданы с точностью до шести знаков после запятой.

Выходные данные

Программа должна вывести единственное целое число.

Примеры**входные данные**

10 30

выходные данные

4

входные данные

10 50

выходные данные

5

Задача №597. Утренняя пробежка - 2

В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал пробег на 70% от предыдущего значения. По данному числу y определите номер дня, на который суммарный пробег спортсмена составит не менее y километров.

Входные данные

На вход программа получает два действительных числа x и y . Числа положительные, действительные, не превосходят 1000, заданы с точностью до шести знаков после запятой.

Выходные данные

Программа должна вывести единственное целое число.

Примеры

входные данные

```
1 20
```

выходные данные

```
6
```

Задача №595. Диета

В некоторой сверхсекретной лаборатории изучаются физические возможности животных. Любой живой организм нуждается в трех компонентах пищи – белках, жирах и углеводах. Известен набор продуктов, имеющийся в распоряжении лаборатории, и меню животных – сколько единиц каждого продукта они получают. Известно также, сколько белков, жиров и углеводов необходимо для нормальной жизнедеятельности животного. Необходимо определить, получает ли животное достаточное количество питательных веществ. Известно, что животному требуется в сутки X белков, Y жиров и Z углеводов.

Известно также, что всего животное получает в сутки N продуктов питания, и для каждого из них известны A_i , B_i , C_i и Q_i – соответственно, энергетическая ценность единицы продукта в белках, жирах и углеводах и количество единиц этого продукта. Все числа – действительные, заданные с точностью до 5 знаков после запятой.

Входные данные

На первой строке входных данных записаны числа X , Y и Z . На второй строке записано число N . Далее на N строках записаны, соответственно, A_i , B_i , C_i и Q_i .

Выходные данные

Выведите YES, если данный пищевой рацион является достаточным по всем параметрам, и NO в противном случае.

Примеры

входные данные

```
1.0 1.0 1.0
3
1 0 0 1
0 0.5 0 2
0 0 0.25 4
```

выходные данные

YES