

Все задания выполняются на одном из предложенных языков: C++, Java, Python.  
Вам нужно прислать ссылку на ваш публичный репозиторий на github.com.

В вашем репозитории должны быть 4 каталога: task1, task2, task3, task4 (Регистр важен), а также файл: author.txt (обязательный), который содержит фамилию\_имя латиницей и имя языка, на котором выполнялось задание [ java | cpp | python ].

\* Для невыполненных заданий каталоги не создаются.

Пример файла author:

```
Ivanov_Ivan  
python
```

В каждом каталоге должны находиться исходные файлы.

Для C++ - task#.cpp, для java - task#.java, для python - task#.py где # - номер задания. В одном каталоге только один файл (Регистр имени важен).

Входные данные считываются из файлов. Имя файла или имя с каталогом файлов с входными данными передается программе через аргументы командной строки. То есть через переменные argv[] – для C++ и args[] – для java и python.

```
$ task1.exe test1.txt  
$ java -jar task1.jar test1.txt  
$ python task1.py input.txt
```

Результат выполнения программы должен выводиться в консоль.

В качестве разделителя нужно использовать символ перехода на новую строку.

Ваша работа будет проверяться автоматически (но код также будет просматриваться).

Если порядок вывода результатов будет нарушен, то решение не будет зачтено.

Обратите внимание, что файлы с входными данными будут наши. Если вы захардкодите путь к своему файлу, то это будет ошибка.

Примечание для C++.

Компилятор g++ -std=c++14.€

Примечание для Python.

Python 3.7.3

Примечание для Java.

Java(TM) SE Runtime Environment (build 1.8.0\_211-b12)

## task1

Напишите программу, которая рассчитывает и подает в стандартный вывод следующие значения:

- 90 перцентиль
- медиана
- максимальное значение
- минимальное значение
- среднее значение

*Входные данные:*

Данные для расчетов считываются из файла, путь к которому подается в виде аргумента.

Числа в файле целые в пределах от -32 768 до 32 767.

Каждое число с новой строки.

В файле не более 1000 строк.

*Вывод:*

Вывод значений в указанной последовательности, каждое значение заканчивается символом новой строки.

Все значения с точностью до сотых: 2.50 2.00 0.03.

### Пример

*Входные данные:*

```
5
8
0
0
1
5
5
5
5
10
6
```

*Вывод:*

```
8.20
5.00
10.00
0.00
4.50
```

## task2

Напишите программу, которая определяет положение точки относительно выпуклого четырехугольника в двумерном пространстве.

Координаты фигуры считываются из файла №1. Это вершины четырехугольника, которые располагаются в порядке обхода фигуры по часовой стрелке. Пример:

```
0 0
0 5.5
5.5 5.5
5.5 0
```

Координаты точек считываются из файла №2. Пример:

```
1.3 1.23
10 9
0 3
5.5 5.5
0 -1
```

Файлы передаются программе в качестве аргументов:

Файл с координатами четырехугольника - 1 аргумент;

Файл с координатами точек - 2 аргумент.

Количество точек от 1 до 100.

Координаты четырехугольника и точек – в диапазоне float.

Вывод каждого положения точки заканчивается символом новой строки.

Соответствия ответов:

0 - точка на одной из вершин

1 - точка на одной из сторон

2 - точка внутри

3 - точка снаружи

### task3

В магазине 5 касс, в каждый момент времени к кассе стоит очередь некоторой длины. Каждые 30 минут измеряется средняя длина очереди в каждую кассу (число с плавающей запятой) и для каждой кассы это значение записывается в соответствующий ей файл (всего 5 файлов).

Каждое значение заканчивается символом новой строки.

Магазин работает 8 часов в день.

Рассматривается только один день.

На момент запуска приложения все значения уже находятся в файлах.

Написать программу, которая по данным замеров определяет интервал времени, когда в магазине было наибольшее количество посетителей за день.

Аргумент программы - путь к каталогу с файлами.

В каталоге будут 5 файлов: Cash1.txt, Cash2.txt ... Cash5.txt (Регистр имени важен!).

Пример одного из файлов:

```
1
1.2
1.3
1.34
1.5
1.9
1.7
1.832
1.91
2.3
2.5
4.9
3.5
2.34
1.242
1.01
```

Выведите номер интервала, в котором было наибольшее число посетителей в очередях магазина на всех кассах.

Первый интервал идет под номером 1, последний под номером 16.

В случае обнаружения нескольких интервалов следует выводить первый из них.

#### task4\*

В течении дня в банк заходят люди, для каждого посещения фиксируется время захода в банк и время выхода.

Банк работает с 8:00 до 20:00.

Написать программу, которая определяет периоды времени, когда в банке было максимальное количество посетителей.

Файл содержит информацию о времени посещения банка каждым посетителем, округленном до минут.

Время входа посетителя меньше либо равно времени выхода.

Выведите интервалы времени, когда в банке было максимальное число посетителей.

Начало и конец интервала разделяются пробелом.

В случае необходимости вывести несколько периодов, в качестве разделителя между ними следует использовать символ перевода строки.

#### Пример

*Входные данные:*

8:00 8:30
8:15 8:45
8:45 9:00
8:30 9:00
9:00 9:30
9:10 9:20

*Вывод:*

8:15 9:00
9:10 9:20