

## Общее описание задач

<b>№</b>	<b>Название задачи</b>	<b>Ограничение на объем используемой памяти</b>	<b>Ограничение на время выполнения программы, секунды</b>	<b>Количество баллов, присвоенных задаче</b>
1.	Калибровка датчиков	$\leq 1\text{Mb}$	$\leq 0,1$	100
2.	Новое расследование Шерлока	$\leq 1\text{Mb}$	$\leq 0,1$	100
3.	Эффективность расчета заработной платы в ИТ-проектах	$\leq 4\text{Mb}$	$\leq 0,2$	100

*Примечание.* При равенстве общего количества очков, приоритет будет отдан участнику, набравшему первым соответствующее количество очков.

## Калибровка датчиков

Научные сотрудники лаборатории ядерных исследований решили провести калибровку датчиков ускорителя частиц. Для этого они фотографируют пролетающие в камере ускорителя частицы. На снимках видны следы частиц, которые описываются функцией  $f(x) = a * x^2 + b * x + c$ , где  $x$  - координата частицы,  $a$ ,  $b$ ,  $c$  - коэффициенты функции. На снимке ученые анализируют след частицы, пролетающей в определенном диапазоне координат  $[M, N]$ . Для калибровки датчиков ученые хотят найти минимум функции  $f(x)$  на отрезке  $[M, N]$  и сравнить это значение с данными, идущими от датчиков. Помогите им сделать это.

**Задание.** Разработайте программу, которая определит минимум функции  $f(x)$  на интервале  $[M, N]$ .

**Входные данные.** Первая строка стандартного ввода будет содержать два целых числа  $N$  и  $M$  - границы интервала  $[M, N]$ . Во второй строке записаны три целых числа  $a$ ,  $b$ ,  $c$  - коэффициенты функции  $f(x)$ .

**Выходные данные.** На первой строке стандартного вывода выведите координаты точки минимума функции  $f(x)$  на интервале  $[M, N]$  с точностью до  $10^{-6}$ .

**Ограничения.**  $-1000 \leq N, M \leq 1000$ ;  $-1000 \leq a, b, c \leq 1000$ . Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл должен называться `calibrare.pas`, `calibrare.c` или `calibrare.cpp`.

### Пример

Ввод	Выход
<pre>-1 2 1 0 0</pre>	<pre>0.000000 0.000000</pre>

Ввод	Выход
<pre>-1 2 1 0 -1</pre>	<pre>0.000000 -1.000000</pre>

Ввод	Выход
<pre>-1 2 1 -2 1</pre>	<pre>1.000000 0.000000</pre>

## Новое расследование Шерлока

В переполненном городе была зарегистрирована кража инновационного устройства. Расследовать это дело поручили детективу Шерлоку, известному умением раскрывать сложнейшие дела. Украденное устройство оснащено трекером, который при активации излучает сигналы, указывающие направление движения вора: Север (N), Восток (E), Юг (S) или Запад (V). Трекер был активирован на расстоянии в момент обнаружения кражи. В ходе преследования детектив Шерлок смог собрать серию из  $m$  сигналов за последние несколько минут, сформировав последовательность подсказок типа «VS?EE??», где вопросительные знаки «?» указывает на искаженные сигналы, направление которых невозможно точно определить. Детектив очень хорошо знает город и знает, что преступник, стремясь избежать быстрой поимки, предпочитает передвигаться исключительно по общественным местам города, чтобы смешаться с толпой. Город можно представить в виде сетки  $k \times l$ , где символ «#» (решетка) обозначает недоступные участки (здания, стены и т. д.), а «.» (точка) открытые общественные места, которые правонарушитель может использовать для передвижения. Пример городского рельефа:

...##...

..#.##..#

..#....##

.##....#..

....#....

Не зная точного места, где находилось украденное устройство в момент подключения трекера, но имея карту города и собранные улики, детектив должен выявить все возможные места, где в текущий момент находится вор с украденным устройством, чтобы организовать эффективный захват и возврат украденного устройства. Один сигнал трекера соответствует перемещению в соседнюю ячейку (вверх - Север, вниз - Юг, влево - Запад, вправо - Восток) в зависимости от направления. Вор не может переместиться в недоступные участки.

**Задание.** Разработать программу, которая будет определять количество возможных текущих локаций, зная карту города и собранные подсказки.

**Входные данные.** Стандартный ввод содержит в первой строке три натуральных числа  $k$ ,  $l$ ,  $m$ , разделённые пробелом. В следующих  $k$  строках находятся  $l$  символов «#» и «.» представляющий карту

города. Последняя строка входных данных описывает сигналы, перехваченные Шерлоком — строку из  $m$  символов, принадлежащих множеству  $\{N, E, S, V, ?\}$ .

**Выходные данные.** Стандартный вывод будет содержать строку, содержащую целое число, обозначающее количество различных текущих мест, где может находиться вор.

**Ограничения.**  $1 \leq k, l \leq 500$ ;  $1 \leq m \leq 5000$ . Исходный набор данных поддерживает только одно решение задачи. Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл должен называться `investigatie.pas`, `investigatie.c` или `investigatie.cpp`.

## Пример

### Ввод

```
3 3 3
..#
#..
...
VEE
```

### Выход

```
1
```

### Ввод

```
4 4 3
..#.
#..#
....
.#.#
SES
```

### Выход

```
2
```

## Пояснения.

В первом примере ход на Запад (V), после которого возможны два хода на Восток (E), возможен только в том случае, если вор находился в позиции (2; 1) на момент подключения трекера. После перемещения VEE вор окажется на позиции (2; 2). Соответственно, возможна только одна позиция вора.

Во втором примере с учетом того, что трекер перехватил движения SES (юг-восток-юг) имеем две возможные ситуации:

- a) Первая возможность отражает ситуацию, когда перехват начался в позиции (0; 1), вор перешел на юг в позицию (1; 1), затем на восток в позицию (1; 2), после чего снова пошел на юг в позицию (2; 2). Таким образом, возможная позиция вора может быть (2; 2).

	0	1	2	3
0	.	.	#	.
1	#	.	.	#
2	.	.	.	.
3	.	#	.	#

- b) Вторая возможность отражает ситуацию, когда перехват начался на позиции (1; 1), вор перешел на юг на позицию (2; 1), затем на восток на позицию (2; 2), после чего снова пошел на юг на позицию (3 ;2). Таким образом, возможная позиция вора может быть (3; 2). Итак, ответ 2 – две возможные позиции вора.

	0	1	2	3
0	.	.	#	.
1	#	.	.	#
2	.	.	.	.
3	.	#	.	#

## Эффективность расчета заработной платы в ИТ-проектах

Михаил является администратором ИТ-компании. Его компания должна завершить  $n$  крупных проектов. По его оценке, для завершения проекта  $i$  требуется  $A_i$  часов работы.

Компания может нанять  $m$  ИТ-специалистов, каждый  $j$  специалист имеет контрактное обязательство отработать максимум  $B_j$  часов. Каждый специалист хочет получить полную зарплату, даже если фактическое количество отработанных часов меньше  $B_j$ . Один специалист может внести вклад в реализацию нескольких проектов, выделяя разное количество часов для каждого проекта, но проект считается завершенным только если к его разработке привлечено как минимум  $k$  специалистов и общее количество отработанных часов в сумме равно  $A_i$ . Вклад каждого специалиста в проект выражается целым числом часов.

Цель Михаила - нанять команду специалистов (не обязательно всех  $m$ ) чтобы обеспечить успешное завершение всех проектов и минимизировать общее количество часов, оплачиваемых без фактической работы. Это требует оптимальной стратегии распределения человеческих ресурсов по проектам с целью максимизации операционной эффективности и минимизации необоснованных затрат.

**Задание.** Разработайте программу, которая определит минимальное количество часов, оплачиваемых дополнительно команде ИТ-специалистов для успешного завершения всех проектов, минимизируя количество необоснованно оплачиваемых часов.

**Входные данные.** Стандартный ввод содержит в первой строке стандартного ввода три натуральных числа  $n, m, k$ , разделённые пробелом. Вторая строка содержит  $n$  целых чисел  $A_i$ , разделённые пробелом. На третьей строке находятся  $m$  целых чисел  $B_j$ , разделённые пробелом.

**Выходные данные.** Стандартный вывод будет содержать строку с общим количеством часов, которые ИТ-специалисты фактически не работают, но за которые компания платит. Если, следуя вышеуказанным правилам, нет никакой возможности завершить предложенные  $n$  проектов, будет отображаться „Imposibil”.

**Ограничения.**  $1 \leq m, n, k, A_i, B_j \leq 300$ . Исходный набор данных поддерживает только одно решение задачи. Ограничения по времени выполнения и объему используемой памяти приведены в общем описании задач, предлагаемых к решению. Исходный файл должен называться `salarizare.pas`, `salarizare.c` или `salarizare.cpp`.

**Пример****Ввод**

```
1 2 2  
6  
4 5
```

**Выход**

```
3
```

**Ввод**

```
1 1 3  
4  
4
```

**Выход**

```
Imposibil
```

**Пояснения.**

В первом примере необходимо завершить один проект и можно нанять максимум два специалиста. Для завершения проекта требуются два ИТ-специалиста. Поскольку для завершения проекта необходимо 6 часов, а специалисты законтрактованы на  $4 + 5$  часов, получается, что 3 часа специалисты не будут работать ( $9 - 6$ ), но зарплату получат.

Во втором примере необходимо завершить один проект и можно нанять одного специалиста. Для завершения проекта требуются три специалиста, но у нас есть только один специалист, поэтому будет невозможно завершить проект.