

## День 1

## Общее описание задач

№	Название задачи	Ограничение на объем используемой памяти	Ограничение на время выполнения программы, секунды	Количество баллов присвоенных задаче
1.	Арена	$\leq 256$ Kb	$\leq 0,002$	100
2.	Бруски	$\leq 4256$ Kb	$\leq 2,700$	100
3.	Звездорыл	$\leq 37$ Mb	$\leq 0,790$	100

*Примечание.* При равенстве общего числа очков, лучшим будет считаться решения с меньшим временем выполнения. В случае равенства и времени выполнения, лучшим будет считаться решение, использующее меньший объем памяти.

### Арена

Первые земляне, прибывшие на Марсе, нашли на этой планете множество старинных руин. В частности, были найдены и руины арен, на которых когда-то, вероятнее всего, боролись марсианские гладиаторы. Каждая из этих арен имеет форму правильного многоугольника<sup>1</sup>, однако размеры и количество сторон различны для разных арен. В каждом углу арены (в каждой вершине многоугольника) был вбит столб. Между столбами натягивались веревки (стороны многоугольника) которые и разграничивали арену.

Недавно археологи с земли обнаружили на Марсе еще одну арену. К несчастью, из всех столбов которые обозначали углы арены, сохранилось только три, остальные были разрушены марсианскими бурями.

В настоящее время археологи пытаются определить, какое минимальное количество столбов  $n$  могли быть использованы для разграничения недавно обнаруженной арены.

**Задание.** Напишите программу, которая, по известным картезианским координатам  $x, y$  каждого из оставшихся трех столбов, определяет минимально возможное количество вершин  $n$  которые могла бы иметь арена.

**Входные данные.** Стандартный ввод содержит три строки. Каждая строка содержит по два вещественных числа, разделённых пробелами.  $i$ -я строка стандартного ввода содержит картезианские координаты  $x_i, y_i$   $i$ -ого столба.

**Выходные данные.** Стандартный вывод содержит в единственной строке целое число  $n$ .

**Ограничения.**  $-1000 \leq x_i, y_i \leq 1000$ ;  $n \leq 100$ . Число десятичных цифр в записи вещественных чисел  $x_i, y_i$  не превышает шести. Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `arena.pas`, `arena.c` или `arena.cpp`.

#### Пример

*Ввод*

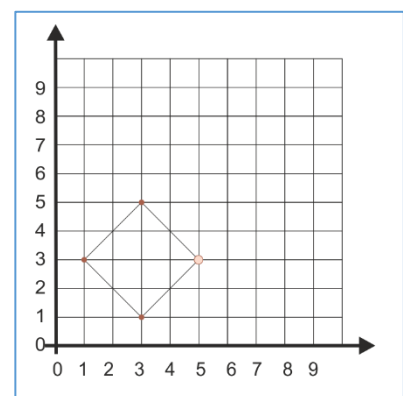
```
1.000000 3.000000
3.000000 1.000000
3.000000 5.000000
```

*Вывод*

```
4
```

#### Объяснение.

Стороны правильного многоугольника имеют одинаковую длину. Так как расстояния между столбами не равны, логично предположить существование еще минимум одного столба (вершины) который использовался для разграничения арены. Действительно, добавление вершины в точку с координатами (5, 3) приводит к образованию квадрата с равными сторонами и конгруэнтными углами. (см. прилагаемый рисунок). В общем случае, количество сторон любого другого правильного многоугольника содержащего исходные точки (столбы), равно  $4k$ , где  $k = 1, 2, 3$  и т.д.



<sup>1</sup> Правильный многоугольник – выпуклый равносторонний многоугольник с конгруэнтными углами.

## Бруски

Рассматривается заготовка в деревянного стержня длиной  $L$ . Из этой заготовки необходимо вырезать бруски меньшей длины, размер каждого из которых выбирается из множества  $\{D_1, D_2, \dots, D_k\}$ .

Очевидно, в общем случае, существует несколько возможных вариантов резки. Они получаются путем выбора определенных размеров из множества  $\{D_1, D_2, \dots, D_k\}$  и установлением чисел подлежащих разрезанию брусков  $n_1, n_2, \dots, n_k$ , в соответствии с выбранными размерами.

Чтобы эффективно использовать сырье, а также не загрязнять окружающую среду, технолог желает выбрать такие размеры и числа подлежащих резке брусков, при которых длина  $R$  оставшейся после резки куска заготовки была минимальной.

Например, для  $L=10, k=2, D_1=2, D_2=4$  минимальное значение  $R=0$  будет получено при использовании одного из следующих методов резки:

- 1) отрезается один брусок размера  $D_1=2$  и два бруска размера  $D_2=4$ ;
- 2) отрезаются пять брусков размера  $D_1=2$ ;
- 3) отрезаются три бруска размера  $D_1=2$  и один брусок размера  $D_2=4$ .

**Задание.** Составьте программу, которая, зная длину заготовки  $L$  и размеры  $\{D_1, D_2, \dots, D_k\}$ , вычисляет минимально возможную длину  $R$  оставшейся части заготовки.

**Входные данные.** Первая строка стандартного ввода содержит целые числа  $L$  и  $k$ , разделенные пробелом. Вторая строка стандартного ввода содержит целые числа  $D_1, D_2, \dots, D_k$ , разделенные пробелами.

**Выходные данные.** Стандартный вывод должен содержать в одной строке целое число  $R$ .

**Ограничения**  $0 < L \leq 4000000$ ;  $k \leq 1000$ ;  $0 < D_i < L, i = 1, 2, \dots, k$ . Длина  $L$  и размеры  $D_i$  являются целыми числами. Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `bare.pas`, `bare.c` или `bare.cpp`.

### Пример 1.

*Ввод*

```
10 3
2 3 4
```

*Вывод*

```
0
```

### Пример 2.

*Ввод*

```
10 2
6 7
```

*Вывод*

```
3
```

### Пример 3

*Ввод*

```
39999 7
30 40 50 60 70 80 90
```

*Вывод*

```
9
```

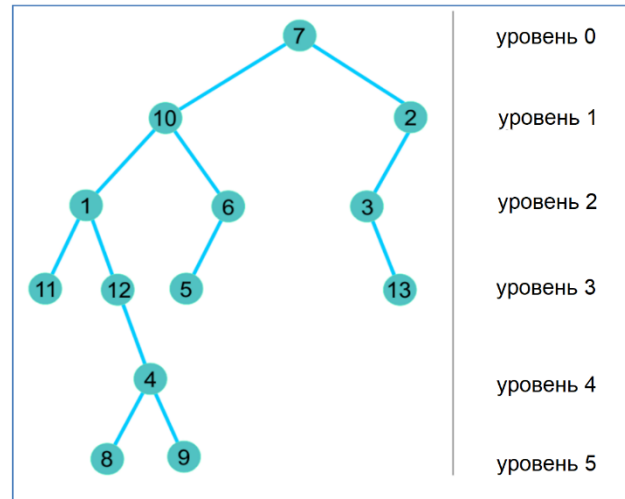
## Звездорыл

У Иона большая проблема – у него на огороде завелся звездорыл. Хотя звездорыл и не вредит, холмики которыми он усеял все грядки портят вид огорода. Ион хотел бы прогнать звездорыла, не навредив ему. Для начала Ион решил разведать жилище звездорыла.

В результате исследований, выполненных с помощью автономного микробота, Ион выяснил, что жилище звездорыла имеет древовидную структуру, с множеством норок (смотри рисунок). Ион пронумеровал норки в произвольном порядке, используя последовательные числа  $1, 2, 3, \dots, N$ .

Звездорыл перемещается между норками по прямым галереям. Входом в жилище звездорыла является ближайшая к поверхности норка (норка находящаяся на уровне 0).

Каждая из находящихся на уровне  $i, i = 1, 2, 3$  и т.д. норок обязательно соединена галереями с норкой находящейся на уровне  $(i-1)$  и с не более чем двумя норками на уровне  $(i+1)$ . Галереи проложены наклонно, следовательно возможны следующие случаи:



- Норка, находящаяся на уровне  $i$ , не имеет галерей к норкам, находящимся на уровне  $(i+1)$ ;
- Норка, находящаяся на уровне  $i$ , соединена галереей с наклоном влево с единственной норкой, находящейся на уровне  $(i+1)$ ;
- Норка, находящаяся на уровне  $i$ , соединена галереей с наклоном вправо с единственной норкой, находящейся на уровне  $(i+1)$ ;
- Норка, находящаяся на уровне  $i$ , соединена наклонными галереями с двумя норками находящимися на уровне  $(i+1)$ : с первой – через галерею с наклоном вправо, со второй – через галерею с наклоном влево.

Чтобы прогнать звездорыла, Ион решил установить в его жилище два отпугивающих устройства – микроскопические ультразвуковые генераторы, излучение которых, как надеется Ион, заставит звездорыла «переселиться» в соседний, пустующий огород.

После долгих размышлений, Ион решил что наилучшим вариантом установки генераторов будет их размещение в крайних, в самой левой и в самой правой норках, находящихся на одном и том же уровне жилища звездорыла. Но, так как нумерация норок производилась произвольно, а данные переданные микроботом имеют специфическую структуру, Ион не знает номера крайних норок выбранного уровня  $K$ .

Например, если бы Ион решил установить отпугивающие устройства в крайние норки, находящихся на 3-м уровне, их номерами были бы 11 и 13.

**Задание.** Напишите программу, которая зная выбранный Ионом уровень  $K$  и данные собранные микро роботом, определяет номера крайних норок, самой левой и самой правой.

**Входные данные.** Первая строка стандартного ввода содержит целые числа  $N$  и  $K$ , разделенные пробелами. Каждая из следующих  $N-1$  строк стандартного ввода содержит по два целых числа и одну из букв S, D, разделенные пробелами. Каждая пара чисел описывает пару норок, соединенных галереями: первое число является номером норки находящейся на уровне  $i$ , второе число является номером норки находящейся на уровне  $(i+1)$ . Буква S указывает на

галерею с наклоном влево (норка находящейся на уровне  $(i+1)$  расположена левее), а буква D указывает на галерею с наклоном вправо (норка находящейся на уровне  $(i+1)$  расположена правее) норки находящейся на уровне  $i$ .

**Выходные данные.** Стандартный вывод содержит в единственной строке два целых числа разделенных пробелом: номер крайней левой норки и номер крайней правой норки заданного уровня  $K$ .

**Ограничения.**  $3 \leq N \leq 1000000$ ;  $0 \leq K \leq 500$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Имя исходного файла `sobol.pas`, `sobol.c` или `sobol.cpp`.

### Пример.

*Ввод*

```
13 3
4 8 S
10 6 D
7 2 D
4 9 D
1 12 D
6 5 S
12 4 D
2 3 S
3 13 D
1 11 S
10 1 S
7 10 S
```

*Вывод*

```
11 13
```