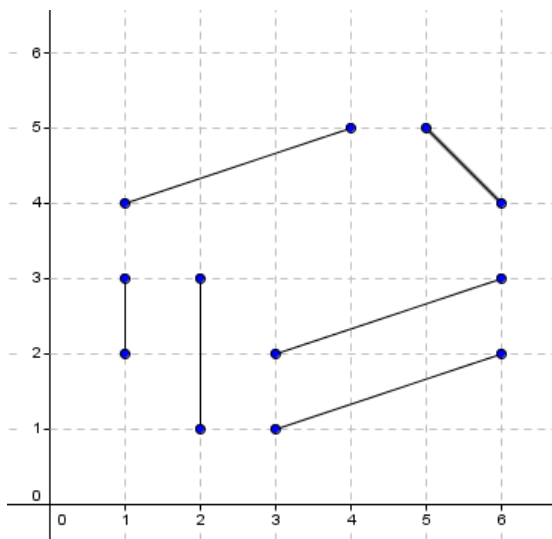


# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг, 16. 12. 2016 г.

Група А (11.-12. клас)

## Задача А1. НАКЛОН



Дадени са няколко отсечки с целочислени координати на крайните си точки. За всяка от отсечките разглеждаме хоризонтална линия, която минава през левия ѝ край (а когато двата края на отсечката лежат на една вертикала, разглеждаме хоризонталната линия, която минава през долния ѝ край). Ъгълът, който сключва отсечката с тази хоризонтална линия, наричаме *наклон* на отсечката. Наклонът се измерва в градуси като реално число и е в диапазона от  $-90$  до  $90$ . Когато отсечката е над разглежданата хоризонтална линия, наклонът е положително число с най-голяма стойност  $90$ . Когато отсечката е под хоризонталната линия, наклонът е отрицателно число, строго по-голямо от  $-90$ . Когато отсечката е хоризонтална, наклонът ѝ е  $0$ .

Някои отсечки могат да имат еднакъв наклон. Напишете програма **slope**, която преброява колко са различните наклони, които имат отсечките от дадена съвкупност.

### Вход

На първия ред на стандартния вход е зададен броят  $n$  на отсечките. Следват  $n$  реда, всеки съдържащ по 4 цели положителни числа, разделени с интервал – координатите  $(x, y)$  на единия край на отсечка и координатите  $(x, y)$  на другия край на същата отсечка.

### Изход

Програмата трябва да извежда на стандартния изход един ред с едно цяло число, равно на търсения брой различни наклони.

### Ограничения

- $0 < n < 100$ ;
- всички координати са цели положителни числа, по-малки от 100. Двата края на всяка отсечка са различни точки.

### Пример

#### Вход

```
6
3 1 6 2
3 2 6 3
1 2 1 3
2 1 2 3
1 4 4 5
6 4 5 5
```

#### Изход

```
3
```

### Обяснение на примера

Данните съответстват на чертежа към задачата.

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг, 16. 12. 2016 г.

Група А (11.-12. клас)

## Задача А2. ОБОБЩЕН ФИБОНАЧИ

Да разгледаме безкрайна редица от цели положителни числа  $\{f_1, f_2, f_3, \dots, f_n, \dots\}$ , определена по следния начин:

- първите две от числата,  $f_1$  и  $f_2$ , са зададени;
- всяко следващо число се получава като сбор от предишните две:  $f_n = f_{n-1} + f_{n-2}$ , за  $n = 3, 4, \dots$

При  $f_1 = 1, f_2 = 1$  това е известната редица на Фибоначи 1, 1, 2, 3, 5, 8, 13, 21, ..., затова, при какви да е зададени начални цели положителни стойности, наричаме редицата *обобщена редица на Фибоначи*.

Да запишем  $n$ -тия член  $f_n$  на тази редица в позиционна бройна система с основа  $p$  (т.е., в  $p$ -ична бройна система). Напишете програма **fib2**, която определя предпоследния символ в този запис. За „цифри“ използваме стандартно приетите символи: 0, 1, 2, ..., 9, A, B, C, D, E, F, съответно със стойности 0, 1, 2, ..., 9, 10, 11, 12, 13, 14, 15. Ако за записа на  $f_n$  се използва само една цифра, приемаме за предпоследен символ 0 („водеща нула“).

### Вход

От стандартния вход се въвеждат:

- ред 1: две цели положителни числа, разделени с интервал и записани десетично: основата  $p$  на разглежданата бройна система и номерът  $n$  на разглеждания член на редицата;
- ред 2: целите положителни числа  $f_1$  и  $f_2$ , записани в  $p$ -ичен запис и разделени с интервал.

### Изход

Програмата трябва да извежда на стандартния изход един ред с един символ: предпоследния символ в  $p$ -ичния запис на  $f_n$ .

### Ограничения

- $2 \leq p \leq 16$ ;
- десетичното цяло положително число  $n$  е записано без водещи нули и с не повече от 100 цифри;
- в записа на всяко от  $p$ -ичните цели положителни числа  $f_1$  и  $f_2$  също няма водещи нули и има не повече от 100 символа;
- в 30% от тестовите примери  $p = 10$  и в записите на  $n, f_1$  и  $f_2$  няма повече от 8 цифри;
- в други 40% от тестовите примери  $n$  се записва с не повече от 18 цифри.

### Пример

#### Вход

```
13 11
1B A
```

#### Изход

```
1
```

### Обяснение на примера

Интересуваме се от предпоследния символ в тринайсетичния запис на единайсетия член от зададената редица. Първите 11 числа в нея са (в тринайсетичен запис): 1B, 0A, 28, 35, 60, 95, 125, 1BA, 312, 4CC, 811 (подчертан е важният предпоследен символ на всеки член).

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг, 16. 12. 2016 г.

Група А (11.-12. клас)

## Задача А3. РЪКОСТИСКАНИЯ

В предприятие работят  $N$  служители, номерирани с числата от 1 до  $N$ . Сутринта на работа първи пристига служител 1, след него служител 2 и т.н., последен идва служител с номер  $N$ . Всеки от тях първо отива при своите приятели (които са дошли преди него на работа) и се ръкостиска с тях. **При всеки такъв поздрав се брой едно ръкостискане за този, който е дошъл по-късно на работа (т.е. за служителя с по-голям номер), но не и за този, който вече е бил там.**

Естествено, ако служител А е приятел на служител В, то и В е приятел на А.

Вие знаете кой служител колко ръкостискания прави сутрин. Напишете програма **hand**, която определя максималния и минималния брой приятели, които е възможно да има някой от служителите.

### Вход

От първия ред на стандартния вход се въвежда естественото число  $N$  – брой на служителите.

От следващия ред се въвеждат  $N$  цели неотрицателни числа, разделени с по един интервал – на позиция  $i$  е броят на ръкостисканията на служител с номер  $i$ .

### Изход

На първия ред на стандартния изход програмата трябва да извежда максималния, а на втория ред – минималния брой възможни приятели от условието на задачата.

### Ограничения

$$2 \leq N \leq 200\,000$$

#### Пример 1

##### Вход

5  
0 0 1 1 1

##### Изход

3  
0

#### Пример 2

##### Вход

5  
0 0 2 2 3

##### Изход

4  
1

### Пояснения

#### Пример 1:

**За максимален брой:** Примерно служители с номера 3, 4 и 5 може да са приятели с номер 2. Тогава номер 2 има трима приятели и няма вариант някой друг да има повече.

**За минимален брой:** В горепосочената ситуация номер 1 няма приятели.

#### Пример 2:

**За максимален брой:** Има варианти, в които служител 3 може да има четирима приятели.

**За минимален брой:** При тези данни няма начин да има служител без нито един приятел. Има вариант с един приятел да е примерно служител 1 или служител 2.