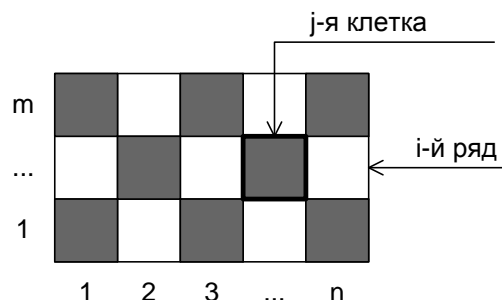


## Задача 5. Шахматная доска

Имя входного файла: `chess.in`  
Имя выходного файла: `chess.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мбайт

Аня разделила доску размера  $m \times n$  на клетки размера  $1 \times 1$  и раскрасила их в черный и белый цвет в шахматном порядке. Васю заинтересовал вопрос: клеток какого цвета получилось больше — черного или белого.

Для того чтобы выяснить это, он спросил у Ани, в какой цвет она раскрасила  $j$ -ю клетку в  $i$ -м ряду доски. По этой информации Вася попытался определить, клеток какого цвета на доске больше.



Требуется написать программу, которая по размерам доски и цвету  $j$ -й клетки в  $i$ -м ряду определит, клеток какого цвета на доске больше — черного или белого.

### Формат входного файла

Входной файл содержит пять целых чисел:  $m$ ,  $n$ ,  $i$ ,  $j$  и  $c$  ( $1 \leq m, n \leq 10^9$ ,  $1 \leq i \leq m$ ,  $1 \leq j \leq n$ ,  $c = 0$  или  $c = 1$ ). Значение  $c = 0$  означает, что  $j$ -я клетка в  $i$ -м ряду доски раскрашена в черный цвет, а значение  $c = 1$  — в белый цвет.

### Формат выходного файла

Выходной файл должен содержать одно из трех слов:

- `black`, если черных клеток на доске больше,
- `white`, если белых клеток на доске больше,
- `equal`, если черных и белых клеток на доске поровну.

### Примеры входных и выходных данных

| <code>chess.in</code> | <code>chess.out</code> |
|-----------------------|------------------------|
| 3 5 1 1 0             | black                  |
| 3 5 2 1 0             | white                  |
| 4 4 1 1 1             | equal                  |

### Примечание

Правильные решения для тестов, в которых  $1 \leq m, n \leq 100$ , оцениваются из 50 баллов.

## Задача 6. Чемпионат по стрельбе

Имя входного файла: shooting.in  
Имя выходного файла: shooting.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мбайт

Победитель школьного этапа олимпиады по информатике нашел дома в старых бумагах результаты чемпионата страны по стрельбе из лука, в котором участвовал его папа. К сожалению, листок с результатами сильно пострадал от времени, и разобрать фамилии участников было невозможно. Остались только набранные каждым участником очки, причем расположились они в том порядке, в котором участники чемпионата выполняли стрельбу.

Распросив папу, школьник выяснил, что количество очков, которое набрал папа, заканчивается на 5, один из победителей чемпионата стрелял раньше, а папин друг, который стрелял сразу после папы, набрал меньше очков. Теперь он заинтересовался, какое самое высокое место мог занять его папа на том чемпионате.

Будем считать, что участник соревнования занял  $k$ -е место, если ровно  $(k - 1)$  участников чемпионата набрали строго больше очков, чем он. При этом победителями считались все участники чемпионата, занявшие первое место.

Требуется написать программу, которая по заданным результатам чемпионата определяет, какое самое высокое место на чемпионате мог занять папа победителя школьного этапа олимпиады по информатике.

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  — количество участников чемпионата страны по стрельбе ( $3 \leq n \leq 10^5$ ).

Вторая строка входного файла содержит  $n$  положительных целых чисел, каждое из которых не превышает 1000, — очки участников чемпионата, приведенные в том порядке, в котором они выполняли стрельбу.

### Формат выходного файла

В выходном файле должно содержаться одно целое число — самое высокое место, которое мог занять папа школьника. Если не существует ни одного участника чемпионата, который удовлетворяет, описанным выше условиям, выведите в выходной файл число 0.

### Примеры входных и выходных данных

| shooting.in             | shooting.out |
|-------------------------|--------------|
| 7<br>10 20 15 10 30 5 1 | 6            |
| 3<br>15 15 10           | 1            |
| 3<br>10 15 20           | 0            |

### Примечание

Правильные решения для тестов, в которых  $1 \leq n \leq 1000$ , оцениваются из 50 баллов.

## Задача 7. Делители

Имя входного файла: `divisors.in`  
Имя выходного файла: `divisors.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мбайт

Натуральное число  $a$  называется делителем натурального числа  $b$ , если  $b = ac$  для некоторого натурального числа  $c$ . Например, делителями числа 6 являются числа 1, 2, 3 и 6. Два числа называются *взаимно простыми*, если у них нет общих делителей кроме 1. Например, 16 и 27 взаимно просты, а 18 и 24 — нет.

Будем называть *нормальным* набор из  $k$  чисел  $(a_1, a_2, \dots, a_k)$ , если выполнены следующие условия:

- 1) каждое из чисел  $a_i$  является делителем числа  $n$ ;
- 2) выполняется неравенство  $a_1 < a_2 < \dots < a_k$ ;
- 3) числа  $a_i$  и  $a_{i+1}$  для всех  $i$  от 1 до  $k - 1$  являются взаимно простыми;
- 4) произведение  $a_1 a_2 \dots a_k$  не превышает  $n$ .

Например, набор (2, 9, 10) является нормальным набором из 3 делителей числа 360.

Требуется написать программу, которая по заданным значениям  $n$  и  $k$  определяет количество нормальных наборов из  $k$  делителей числа  $n$ .

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  и  $k$  ( $2 \leq n \leq 10^8$ ,  $2 \leq k \leq 10$ ).

### Формат выходного файла

В выходном файле должно содержаться одно число — количество нормальных наборов из  $k$  делителей числа  $n$ .

### Пример входных и выходных данных

| <code>divisors.in</code> | <code>divisors.out</code> |
|--------------------------|---------------------------|
| 90 3                     | 16                        |
| 10 2                     | 4                         |

### Примечание

Правильные решения для тестов, в которых  $n \leq 1000$  и  $k = 2$ , оцениваются из 30 баллов.

Правильные решения для тестов, в которых  $k = 2$ , оцениваются из 60 баллов (в эти баллы включаются также 30 баллов для случая  $n \leq 1000$ ,  $k = 2$ ).

## Задача 8. Дом у дороги

Имя входного файла: house.in  
Имя выходного файла: house.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 Мбайт

Министерство дорожного транспорта решило построить себе новый офис. Поскольку министр регулярно выезжает с инспекцией наиболее важных трасс, было решено, что офис министерства не должен располагаться слишком далеко от них.

Наиболее важные трассы представляют собой прямые на плоскости. Министерство хочет выбрать такое расположение для своего офиса, чтобы максимум из расстояний от офиса до трасс был как можно меньше.

Требуется написать программу, которая по заданному расположению наиболее важных трасс определяет оптимальное расположение дома для офиса министерства дорожного транспорта.

### Формат входного файла

Первая строка входного файла содержит одно целое число  $n$  — количество наиболее важных трасс ( $1 \leq n \leq 10^4$ ).

Последующие  $n$  строк описывают трассы. Каждая трасса описывается четырьмя целыми числами  $x_1, y_1, x_2$  и  $y_2$  и представляет собой прямую, проходящую через точки  $(x_1, y_1)$  и  $(x_2, y_2)$ . Координаты заданных точек не превышают по модулю  $10^4$ . Точки  $(x_1, y_1)$  и  $(x_2, y_2)$  ни для какой прямой не совпадают.

### Формат выходного файла

Выходной файл должен содержать два разделенных пробелом вещественных числа: координаты точки, в которой следует построить офис министерства дорожного транспорта. Координаты по модулю не должны превышать  $10^9$ , гарантируется, что хотя бы один такой ответ существует. Если оптимальных ответов несколько, необходимо вывести любой из них.

Ответ должен иметь абсолютную или относительную погрешность не более  $10^{-6}$ , что означает следующее. Пусть максимальное расстояние от выведенной точки до некоторой трассы равно  $x$ , а в правильном ответе оно равно  $y$ . Ответ будет засчитан, если значение выражения  $|x - y| / \max\{1, |y|\}$  не превышает  $10^{-6}$ .

### Примеры входных и выходных данных

| house.in | house.out |
|----------|-----------|
| 4        | 0.5 0.5   |
| 0 0 0 1  |           |
| 0 0 1 0  |           |
| 1 1 2 1  |           |
| 1 1 1 2  |           |

### Примечание

Правильные решения для тестов, в которых  $n \leq 100$  и все прямые параллельны, оцениваются из 20 баллов.

Правильные решения для тестов, в которых  $n \leq 100$  и все прямые параллельны осям координат, оцениваются из 20 баллов.

Правильные решения для тестов, в которых  $n \leq 100$ , оцениваются из 70 баллов (в эти баллы включаются также по 20 баллов за случаи, описанные в предыдущих двух абзацах).