

Задача А. Максимальный элемент

Ограничение времени: 1 с
Ограничение памяти: 512М

Никита начал изучать программирование совсем недавно. Все, что он научился делать на сегодняшний день - это находить максимальный элемент в массиве. Чтобы закрепить этот полезный навык, Никита решил попрактиковаться. Для этого он взял массив A из N целых чисел и выполнил с ним ровно K преобразований.

Каждое преобразование заключалось в том, что Никита находил в массиве максимальный элемент M и заменял каждый элемент массива A_i значением $M - A_i$.

Вам будет задан массив, который был у Никиты в самом начале. Выведите массив, который у него получился после выполнения заданного числа преобразований.

Формат входных данных

В первой строке входных данных записаны два натуральных числа N ($1 \leq N \leq 10^5$) и K ($1 \leq K \leq 10^5$) - количество элементов в исходном массиве и количество преобразований, которое должен выполнить Никита.

Во второй строке входных данных записаны N целых чисел, разделенные пробелами - элементы A_i исходного массива, по абсолютной величине не превышающие $2 \cdot 10^9$.

Формат результата

Выведите массив, который получился у Никиты в результате выполнения K описанных в условии преобразований. Массив необходимо вывести в одной строке. Элементы массива должны быть разделены одним пробелом.

Примеры

Входные данные

```
5 1
3 2 -1 5 4
```

Результат работы

```
2 3 6 0 1
```

Входные данные

```
5 2
13 -9 -3 16 11
```

Результат работы

```
22 0 6 25 20
```

Задача В. Ближайший квадрат

Ограничение времени: 1 с
Ограничение памяти: 512М

Вам будет задано натуральное число N . Найдите самое маленькое натуральное число X большее N , которое является полным квадратом. Натуральное число X называется полным квадратом, если существует такое натуральное число K , что $K^2 = X$.

Формат входных данных

Входные данные содержат единственное натуральное число N ($1 \leq N \leq 10^9$).

Формат результата

Выведите единственное натуральное число - минимально возможное X такое, что $X > N$ и X является полным квадратом.

Примеры

Входные данные

10

Результат работы

16

Входные данные

20

Результат работы

25

Входные данные

48

Результат работы

49

Задача С. Красивая строка

Ограничение времени: 1 с

Ограничение памяти: 512М

Дана строка A , состоящая только из маленьких букв английского алфавита. Построим из нее новую строку B , длиной ровно N символов. Возьмем для этого первоначально пустой массив символов длиной N , заполненный пробелами. Будем выполнять следующие шаги, пока массив B не заполнится буквами:

1. Находим в массиве B самый длинный подмассив подряд идущих ячеек, не заполненных буквами. Если таких подмассивов несколько, выбираем самый левый из них.
2. Вычисляем индекс среднего элемента этого подмассива, округляя, если необходимо, значение в меньшую сторону, и записываем в ячейку массива B с этим индексом очередную букву из строки A . Если мы дошли до конца строки A , то переходим на ее начало и продолжаем брать буквы для строки B от начала строки A .
3. Если в массиве B еще остались незаполненные буквами ячейки, переходим к шагу 1.

После того, как массив B будет весь заполнен буквами, мы получим искомую строку B .

Рассмотрим пример. Пусть строка $A = "abc"$. Построим строку B длиной 10 символов. Первоначально $B = "_____"$, здесь для наглядности символом $_$ обозначен пробел. В соответствии с шагом 1 самый длинный подмассив, не содержащий ни одной буквы, это сам массив B . Пусть его элементы проиндексированы, начиная с 0. Тогда индекс среднего элемента будет 4. Записываем в ячейку массива B с индексом 4 первую букву строки A (буква a). Получаем строку

$B = "_____a_____"$.

Так как массив B еще не заполнен буквами, находим в нем самый длинный подмассив, не содержащий ни одной буквы. Это подмассив, который начинается в ячейке с индексом 5 и заканчивается в ячейке с индексом 9. Его средний элемент имеет индекс 7. Записываем в ячейку с этим индексом следующую букву массива A (буква b). Получаем

$B = "_____a__b____"$.

Теперь самый большой подмассив, не содержащий ни одной буквы, начинается в ячейке с индексом 0 и заканчивается в ячейке с индексом 3. Его середина будет в ячейке с индексом 1. Записываем в нее следующую букву строки A (буква c). Получим

$B = "_c__a__b____"$.

Теперь в массиве B есть три подмассива длины 2, не содержащие букв. Берем самый левый из них и записываем туда очередную букву строки A . Так как на предыдущем шаге мы дошли до конца строки A , то берем опять первую букву строки A (буква a). Получим

$B = "_ca_a__b____"$.

Выполняя описанные в условии шаги, получим далее такую последовательность заполнения массива B :

`V = "_ca_ab_b_"`

`V = "_ca_ab_bc_"`

`V = "aca_ab_bc_"`

`V = "acabab_bc_"`

`V = "acababcbc_"`

`V = "acababcbca"`

Напишите программу, которая по заданной строке *A*, заданной длине строки *V* и по заданному индексу *K* определит, какая буква находится в позиции с индексом *K* в строке *V*. Элементы строки *V* проиндексированы с нуля.

Формат входных данных

В первой строке входных данных задана строка *A*, состоящая только из маленьких букв английского алфавита. Длина строки *A* не превышает 100 символов.

Во второй строке входных данных записаны через пробел два натуральных числа N ($1 \leq N \leq 10^6$) и K ($0 \leq K < N$) - количество символов в строке *V* и индекс буквы в строке *V*, которую нужно вывести.

Формат результата

Выведите букву, которая находится в строке *V* в позиции с индексом *K*.

Примеры

Входные данные

```
abc
10 7
```

Результат работы

```
b
```

Входные данные

```
abc
10 0
```

Результат работы

```
a
```

Входные данные

```
ab
6 3
```

Результат работы

```
a
```

Задача D. Увеличение войска

Ограничение времени: 1 с

Ограничение памяти: 512M

Урфин Джюс решил увеличить численность своего войска. Сейчас в его войске S деревянных солдат, а Урфин хочет, чтобы их стало T ($S \leq T$). Сделать это Урфин Джюс хочет как можно скорее, поэтому он попросил волшебника Изумрудного города Гудвина, великого и ужасного, показать всем, какой он волшебник на самом деле, и увеличить численность войска Урфина Джюса.

Гудвин умеет делать с численностью войска дуболомов всего две операции - он может либо увеличить численность войска на A дуболомов, либо увеличить численность войска ровно в M раз. Делать эти операции он может сколько угодно раз в любой последовательности. Если Гудвин хочет получить из численности S численность T , то он должен начать с числа S и, применяя к получающимся числам указанные выше две операции, получить последовательность чисел, заканчивающуюся числом T .

Рассмотрим пример. Пусть $S = 1$, $T = 20$, $A = 1$, $M = 2$. Гудвин должен получить число 20 из числа 1 либо прибавляя к получающимся числам 1, либо умножая их на 2. Одна из возможных последовательностей, переводящих 1 в 20 с помощью описанных выше операций, выглядит так: 1->2->4->8->16->17->18->19->20. Длина этой цепочки равна 9.

Гудвин хочет знать можно ли получить число T из числа S , если да, то сколькими способами можно это сделать, и какова длина самой короткой последовательности чисел, ведущей из S в T (включая сами числа S и T).

Формат входных данных

В единственной строке входных данных записаны четыре натуральных числа, разделенные пробелами, S , T , A , M ($1 \leq S \leq T \leq 10^6$, $1 \leq A \leq 100$, $2 \leq M \leq 100$).

Формат результата

Если получить число T из числа S невозможно, выведите NO.

Иначе выведите два числа, разделенные пробелом, - количество способов получить число T из числа S и длину самой короткой последовательности чисел, ведущей от числа S до числа T . Входные данные будут таковы, что количество способов не будет превышать 10^{18} .

Примеры

Входные данные

1 4 1 2

Результат работы

4 3

Входные данные

1 20 1 2

Результат работы

60 6

Входные данные

3 6 2 3

Результат работы

NO

Задача Е. Платформы

Ограничение времени: 1 с

Ограничение памяти: 512М

Никита создаёт компьютерную игру. Сейчас он проектирует один из уровней с бонусами. Этот уровень состоит из множества платформ, висящих в воздухе, на которых могут находиться одна или несколько монет. Персонаж игрока может перемещаться по этим платформам. При выходе за пределы платформы, он начинает падать вниз, пока не попадёт на ближайшую снизу платформу или, если такой нет, пока не вылетит за пределы уровня. Когда персонаж попадает на некоторую платформу, он автоматически подбирает все монеты, которые на ней лежат.

Чтобы описать уровень, введём систему координат следующим образом: ось OX направим слева направо, а ось OY вертикально вверх. Будем считать, платформы отрезками, параллельными оси OX . Концами платформ будут точки (l_i, h_i) и (r_i, h_i) . Все платформы расположены на разных высотах. Если игрок начал падать из точки (x, y) , то он окажется на самой высокой платформе ниже начальной, которая включает в себя точку с абсциссой x . Платформа включает в себя самую левую и самую правую точки, а также всё, что между ними.

Уровень заканчивается, когда персонаж вылетает за пределы уровня. Игрок начинает уровень на самой верхней платформе. Какое максимальное количество монет может получить игрок?

Формат входных данных

Первая строка содержит одно число N ($1 \leq N \leq 5000$) — количество платформ.

Следующие N строк содержат по четыре числа l_i, r_i, h_i, g_i ($1 \leq l_i < r_i \leq 10^6, 0 \leq h_i, g_i \leq 10^6$) — самая левая и самая правая границы платформы, высота, на которой она находится, и количество монет на платформе. Гарантируется, что $h_i > h_{i+1}$.

Формат результата

Выведите одно число — максимальное количество монет, которые может собрать игрок.

Примеры

Входные данные

```
5
6 10 7 1
9 11 5 5
2 6 4 6
2 4 2 4
1 15 1 0
```

Результат работы

```
11
```

Входные данные

```
5
```

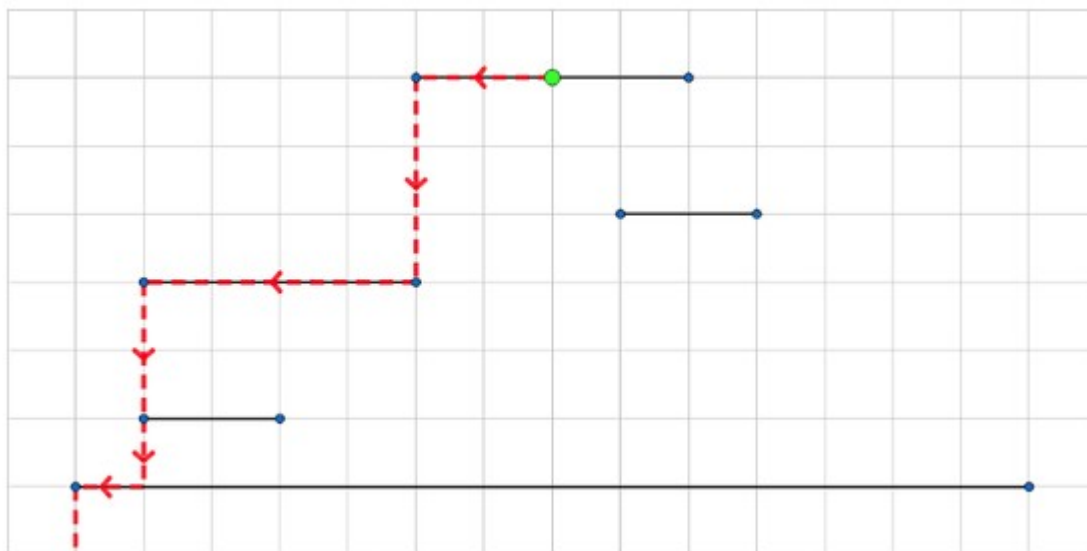
6 10 7 1
9 11 5 7
2 6 4 2
2 4 2 4
1 15 1 0

Результат работы

8

Примечания

Оптимальный путь в первом тесте, который соберёт монеты со всех платформ, кроме платформы на высоте 5, выглядит следующим образом:



Задача F. Кодовый замок

Ограничение времени: 1 с

Ограничение памяти: 512M

Кодовый замок

Кодовый замок представляет собой матрицу чисел размера $N \times M$, в каждой строке которой первоначально записаны последовательные числа от 1 до M . Например, при $N = 3$ и $M = 7$ замок будет иметь вид:

```
1234567
1234567
1234567
```

Чтобы закрыть замок, необходимо циклически сдвинуть некоторые строки матрицы на какое-нибудь количество клеток влево или вправо. Чтобы открыть замок, необходимо циклически сдвинуть некоторые строки матрицы вправо или влево так, чтобы в каком-нибудь столбце оказалась кодовая комбинация чисел.

Например, закроем замок, сдвинув циклически первую строку на 2 позиции влево, вторую строку на 3 позиции вправо, а третью строку на 1 позицию вправо. Получим закрытый замок:

```
3456712
5671234
7123456
```

Пусть кодовая комбинация для открытия замка - 621. Тогда замок можно открыть, например, так: первую строку не сдвигать никуда, вторую строку сдвинуть на 1 позицию влево, а третью строку сдвинуть на 2 позиции вправо. Получим открытый замок:

```
3456712
6712345
7561234
```

Как видно, теперь в четвертом столбце появилась кодовая комбинация 621, и замок считается открытым. В этом примере, чтобы открыть замок нам пришлось сдвинуть строки в сумме на 3 позиции - на 1 позицию вторую строку, и на 2 позиции третью строку. Замок можно было бы открыть и по другому. Например, можно было бы сдвинуть первую строку на 1 позицию вправо, вторую строку не сдвигать, а третью строку сдвинуть на 3 позиции вправо. Тогда бы мы получили замок вида:

```
2345671
5671234
4567123
```

При этом кодовая комбинация оказалась бы в пятом столбце, а нам пришлось бы сдвигать строки массива в сумме на 4 позиции. В первом варианте суммарное количество позиций, на которое надо сдвинуть строки оказалось меньше. Очевидно, что есть и другие способы открыть замок с другим суммарным количеством позиций, на которой будут сдвигаться строки.

Напишите программу, которая по заданному закрытому замку и по заданной кодовой комбинации определит минимально возможное суммарное количество позиций, на которое придется сдвинуть строки замка, чтобы открыть его.

Формат входных данных

В первой строке входных данных записаны два числа, разделенные пробелом, N и M ($1 \leq N, M \leq 1000$) - количество строк и столбцов в замке.

В каждой из последующих N строк записан некоторый циклический сдвиг чисел $1, 2, 3, \dots, M$. Числа в каждой строке разделены пробелами.

В последней строке входных данных записаны N чисел, разделенных пробелами - кодовая комбинация для открытия замка.

Формат результата

Выведите единственное число - минимально возможное суммарное количество позиций, на которое придется сдвинуть строки замка, чтобы в каком-то столбце замка получить кодовую комбинацию. Входные данные будут таковы, что с помощью заданной кодовой комбинации замок обязательно можно открыть.

Примеры

Входные данные

```
3 7
3 4 5 6 7 1 2
5 6 7 1 2 3 4
7 1 2 3 4 5 6
6 2 1
```

Результат работы

3

Входные данные

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

Результат работы

4

Входные данные

```
3 3
1 2 3
2 3 1
3 1 2
2 3 1
```

Результат работы

0

Задача G. Квадраты

Ограничение времени: 1 с

Ограничение памяти: 512М

На плоскости задано множество различных точек с целочисленными координатами. Посчитайте, сколько существует квадратов с вершинами в этих точках и сторонами, параллельными осям координат.

Формат входных данных

Первая строка входных данных содержит целое число N ($4 \leq N \leq 10^3$) - количество точек.

В каждой из последующих N строк записаны два целых числа X_i и Y_i ($-10^9 \leq X_i, Y_i \leq 10^9$) - координаты точек. Гарантируется, что все точки во входных данных будут различны.

Формат результата

Выведите единственное число - количество квадратов со сторонами, параллельными осям координат, вершины которых лежат в точках из заданного множества.

Примеры

Входные данные

```
4
0 0
3 3
3 0
0 3
```

Результат работы

```
1
```

Входные данные

```
9
4 2
0 0
0 4
0 2
2 2
4 0
2 0
2 4
4 4
```

Результат работы

```
5
```

Первая олимпиада школьников Крыма по программированию
Финальный тур
21 марта 2020 г.

Входные данные

4
0 2
2 0
0 -2
-2 0

Результат работы

0

Задача Н. Погрузчик

Ограничение времени: 1 с

Ограничение памяти: 512М

Склад представляет собой матрицу из N строк и M столбцов. Пусть строки матрицы пронумерованы от 1 до N , а столбцы от 1 до M . В ячейке матрицы, расположенной в i -ой строке и j -ом столбце, находится груз весом A_{ij} .

Пусть строки матрицы пронумерованы от 1 до N , а столбцы от 1 до M . Погрузчик собирает грузы за несколько проходов по матрице:

- Погрузчик начинает движение из клетки (1,1) и совершает "круг" по внешним краям матрицы, то есть он сначала движется по первой строке слева направо, затем по последнему столбцу сверху вниз, затем по последней строке справа налево, затем по первому столбцу снизу вверх. Во время своего движения погрузчик собирает все грузы из соответствующих клеток. Когда проход завершен, кладовщик записывает суммарный вес собранных грузов. После этого погрузчик освобождается и идет на второй проход.
- Погрузчик начинает движение из клетки (2,2) и совершает "круг" по строкам и столбцам, удаленным от края матрицы на 1, то есть от сначала движется по второй строке слева направо, затем по предпоследнему столбцу сверху вниз, затем по предпоследней строке справа налево, затем по второму столбцу снизу вверх. Точно так же, во время своего движения погрузчик собирает все грузы из соответствующих клеток. Когда второй проход завершен, кладовщик опять записывает суммарный вес собранных грузов. Погрузчик освобождается и идет на следующий проход.
- И так далее. Проходы повторяются до тех пор, пока не будут собраны все грузы.

Например, пусть исходная матрица имеет вид

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

Тогда погрузчик соберет все грузы за два прохода. После первого прохода кладовщик запишет, что суммарный вес собранных грузов равен $3+2+1+5+5+1+6+5+3+4+6+4 = 45$. После второго прохода кладовщик запишет вес $14 = 3+2+4+5$.

Напишите программу, которая по заданной матрице определит максимальное число, которое будет записано кладовщиком, то есть максимальный суммарный вес грузов, который погрузчик соберет во время одного из своих проходов.

Формат входных данных

В первой строке входных данных записаны два целых числа N и M ($1 \leq N, M \leq 100$) - размеры матрицы.

В каждой из последующих N строк записаны через пробел M чисел - веса грузов A_{ij} ($1 \leq A_{ij} \leq 1000$).

Формат результата

Выведите единственное число - максимальный суммарный вес грузов, который погрузчик может собрать во время одного из своих проходов.

Примеры

Входные данные

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

Результат работы

45

Входные данные

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

Результат работы

27

Входные данные

```
3 5
1 2 1 1 2
1 5 6 9 1
1 2 1 1 2
```

Результат работы

20

Задача I. Кирпичи

Ограничение времени: 1 с

Ограничение памяти: 512М

К кирпичей весят N фунтов и еще полкирпича. Сколько весит один кирпич?

Формат входных данных

Входные данные содержат два числа, разделенные пробелом, - натуральное число K ($1 \leq K \leq 1000$), и положительное действительное число N , по величине не превышающее 10^6 .

Формат результата

Выведите вес кирпича, округленный ровно до 6 знаков после десятичной точки.

Примеры

Входные данные

1 1.000

Результат работы

2.000000

Входные данные

2 1.000

Результат работы

0.666667

Входные данные

3 1.000

Результат работы

0.400000

Задача J. Подстроки

Ограничение времени: 1 с

Ограничение памяти: 512М

Задана строка, состоящая только из букв a, b, c латинского алфавита. Подсчитайте, сколько в этой строке подстрок, либо начинающихся с буквы a, и заканчивающихся буквой b, либо начинающихся с буквы b и заканчивающихся буквой c.

Формат входных данных

Входные данные содержат единственную строку, состоящую только из букв a, b, c латинского алфавита. Длина строки не менее 3 и не более 10^6 символов.

Формат результата

Выведите единственное число - количество подстрок, удовлетворяющих условию задачи.

Примеры

Входные данные

abc

Результат работы

2

Входные данные

aabbcc

Результат работы

8

Входные данные

abacabacaba

Результат работы

12