**Задача 1**

Выведите все элементы списка с четными индексами (то есть A[0], A[2], A[4], ...).

Программа должна быть эффективной и не выполнять лишних действий!

**Входные данные**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**Входные данные**

1 2 3 4 5

**Выходные данные**

1 3 5

**Задача 2**

Выведите все четные элементы списка.

**Входные данные**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

Входные данные

1 2 2 3 3 3 4

**Выходные данные**

2 2 4

**Задача 3**

Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет - не выводите ничего. Если таких пар соседей несколько - выведите первую пару.

**Входные данные**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**Входные данные**

-1 2 3 -1 -2

**Выходные данные**

2 3

**Задача 4**

Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать.

**Входные данные**

Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200.

**Выходные данные**

Выведите номер, под которым Петя должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.

**Примеры**

**Входные данные**

165 163 160 160 157 157 155 154

162

**Выходные данные**

3

**Входные данные**

165 163 160 160 157 157 155 154

160

**Выходные данные**

5

**Задача 5**

Дан список, упорядоченный по неубыванию элементов в нем. Определите, сколько в нем различных элементов.

**Входные данные**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**Входные данные**

1 2 2 3 3 3

**Выходные данные**

3

**Подсказка:** чтобы добавить элемент в лист можно использовать функцию append:

a = [1,2,3]

a.append(4) # теперь a == [1,2,3,4]

**Подсказка 2:** а ещё в этой задаче вы можете использовать словари. Для этого вам нужно знать как проверить, есть ли определённый ключ в словаре:

a = {‘k’: 0}

‘k’ in a # True

‘x’ in a # True

**Задача 6**

В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

**Дан массив чисел**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**Входные данные**

3 4 5 2 1

**Выходные данные**

3 4 1 2 5

**Задача 7**

Дан список. Выведите те его элементы, которые встречаются в списке только один раз. Элементы нужно выводить в том порядке, в котором они встречаются в списке.

**Входные данные**

Дан массив чисел

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**Входные данные**

1 2 2 3 3 3

**Выходные данные**

1

**Задача 8**

N кеглей выставили в один ряд, занумеровав их слева направо числами от 1 до N. Затем по этому ряду бросили K шаров, при этом i-й шар сбил все кегли с номерами от li до ri включительно. Определите, какие кегли остались стоять на месте.

**Входные данные**

Программа получает на вход количество кеглей N и количество бросков K. Далее идет K пар чисел li, ri, при этом 1https://informatics.msk.ru/moodle/lib/jsMath/fonts/cmsy10/alpha/120/char14.pnglihttps://informatics.msk.ru/moodle/lib/jsMath/fonts/cmsy10/alpha/120/char14.pngrihttps://informatics.msk.ru/moodle/lib/jsMath/fonts/cmsy10/alpha/120/char14.pngN.

**Выходные данные**

Программа должна вывести последовательность из N символов, где j-й символ есть “I”, если j-я кегля осталась стоять, или “.”, если j-я кегля была сбита.

**Примеры**

**Входные данные**

10 3 #N, K

[[8, 10],

[2 5],

[3 6]]

**Выходные данные**

I.....I...

**Задача 9**

Известно, что на доске 8×8 можно расставить 8 ферзей так, чтобы они не били друг друга. Вам дана расстановка 8 ферзей на доске, определите, есть ли среди них пара бьющих друг друга.

**Входные данные**

Программа получает на вход восемь пар чисел, каждое число от 1 до 8 - координаты 8 ферзей.

**Выходные данные**

Если ферзи не бьют друг друга, выведите слово NO, иначе выведите YES.

**Примеры**

**Входные данные**

1 7

2 4

3 2

4 8

5 6

6 1

7 3

8 5

**Выходные данные**

NO

**Входные данные**

1 8

2 7

3 6

4 5

5 4

6 3

7 2

8 1

**Выходные данные**

YES

**Задача 10**

Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

**Входные данные**

Программа получает на вход число n https://informatics.msk.ru/moodle/lib/jsMath/fonts/cmsy10/alpha/120/char14.png100, являющееся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по n чисел, являющихся элементами массива.

**Выходные данные**

Программа должна выводить слово yes для симметричного массива и слово no для несимметричного.

**Примеры**

**Входные данные**

3 # n

0 1 2

1 5 3

2 3 4

**Выходные данные**

yes

**Входные данные**

3 #n n

0 0 0

0 0 0

1 0 0

**Выходные данные**

no

**Задача 11**

В метании молота состязается n спортcменов. Каждый из них сделал m бросков. Победителем считается тот спортсмен, у которого сумма результатов по всем броскам максимальна.   
Если перенумеровать спортсменов числами от 0 до n-1, а попытки каждого из них – от 0 до m-1, то на вход программа получает массив A[n][m], состоящий из неотрицательных целых чисел. Программа должна определить максимальную сумму чисел в одной строке и вывести на экран эту сумму и номер строки, для которой достигается эта сумма.

**Входные данные**

Программа получает на вход два числа n и m, являющиеся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по m чисел, являющихся элементами массива.

**Выходные данные**

Программа должна вывести  2 числа: сумму и номер строки, для которой эта сумма достигается. Если таких строк несколько, то выводится номер наименьшей из них. Не забудьте, что нумерация строк (спортсменов) начинается с 0.

**Примеры**

**Входные данные**

2 2

5 4

3 5

**Выходные данные**

9

0

**Задача 12**

В метании молота состязается n спортcменов. Каждый из них сделал m бросков. Победителем соревнований объявляется тот спортсмен, у которого максимален наилучший результат по всем броскам. Таким образом, программа должна найти значение максимального элемента в данном массиве, а также его индексы (то есть номер спортсмена и номер попытки).

**Входные данные**

Программа получает на вход два числа n и m, являющиеся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по m чисел, являющихся элементами массива.

**Выходные данные**

Программа выводит значение максимального элемента, затем номер строки и номер столбца, в котором он встречается. Если в массиве несколько максимальных элементов, то нужно вывести минимальный номер строки, в которой встречается такой элемент, а если в этой строке таких элементов несколько, то нужно вывести минимальный номер столбца. Не забудьте, что все строки и столбцы нумеруются с 0.

**Примеры**

**Входные данные**

3 3 #n, m

3 1 2

1 3 4

3 3 3

**Выходные данные**

4

1 2

**Задача 13**

В метании молота состязается n спортcменов. Каждый из них сделал m бросков. Побеждает спортсмен, у которого максимален наилучший бросок. Если таких несколько, то из них побеждает тот, у которого наилучшая сумма результатов по всем попыткам. Если и таких несколько, победителем считается спортсмен с минимальным номером. Определите номер победителя соревнований.

**Входные данные**

Программа получает на вход два числа n и m, являющиеся числом строк и столбцов в массиве. Далее во входном потоке идет n строк по m чисел, являющихся элементами массива.

**Выходные данные**

Программа должна вывести одно число - номер победителя соревнований. Не забудьте, что  строки  (спортсмены) нумеруются с 0.

**Примеры**

**Входные данные**

3 3 # n, m

1 2 7

1 3 5

4 1 6

**Выходные данные**

0

**Задача 14**

Напишите программу, отображающую игровое поле для игры "Сапер".

**Входные данные**

Даны числа *N* и *M* (целые, положительные, не превышают 32) – количество строк и столбцов в поле соответственно, далее число *W* (целое, неотрицательное, не больше 1000) – количество мин на поле, далее следует *W* пар чисел, координаты мины на поле (первое число – строка, второе число – столбец).

**Выходные данные**

Требуется вывести на экран поле. Формат вывода указан в примере.

**Примеры**

**Входные данные**

3 2 # N, M

2 # W

1 1

2 2

**Выходные данные**

\* 2

2 \*

1 1

**Входные данные**

2 2 # N, M

0 # W

**Выходные данные**

0 0

0 0