

Задания #21

#1. Вводится натуральное число N (то есть, положительное, целое). Требуется создать двумерный (вложенный) список размером $N \times N$ элементов, состоящий из всех единиц, а затем, в последний столбец записать пятерки. Вывести этот список на экран в виде таблицы чисел, как показано в примере ниже.

P.S. Будьте внимательны в конце строк пробелов быть не должно!

Sample Input:

4

Sample Output:

```
1 1 1 5
1 1 1 5
1 1 1 5
1 1 1 5
```

#2. Вводится список из URL-адресов (каждый с новой строки). Требуется в них заменить все пробелы на символ дефиса (-). Следует учесть, что может быть несколько подряд идущих пробелов. Результат преобразования вывести на экран в виде строк из URL-адресов.

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

Sample Input:

```
django chto eto takoe poryadok ustanovki
model mtv marshrutizaciya funkcii predstavleniya
marshrutizaciya obrabotka isklyucheniyy zaprosov
perenapravleniya
```

Sample Output:

```
django-chto-eto-takoe-poryadok-ustanovki
model-mtv-marshrutizaciya-funkcii-predstavleniya
marshrutizaciya-obrabotka-isklyucheniyy-zaprosov-
perenapravleniya
```

```
import sys
```

```
lst_in = list(map(str.strip, sys.stdin.readlines()))
```

#3. Вводится натуральное число n . Необходимо найти все простые числа, которые меньше этого числа n , то есть, в диапазоне $[2; n)$. Результат вывести на экран в строчку через пробел.

Sample Input:

11

Sample Output:

2 3 5 7

#4. Вводится двумерный список размерностью 5 x 5 элементов, состоящий из нулей и, в некоторых позициях, единиц (см. пример ввода ниже). Требуется проверить, не касаются ли единицы друг друга по горизонтали, вертикали и диагонали. То есть, вокруг каждой единицы должны быть нули. Если проверка проходит вывести ДА, иначе - НЕТ.

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

Sample Input:

```
1 0 0 0 0
0 0 1 0 1
0 0 0 0 0
0 1 0 1 0
0 0 0 0 0
```

Sample Output:

ДА

```
import sys

import numpy as np

# считывание списка из входного потока

s = sys.stdin.readlines()

lst_in = [list(map(int, x.strip().split())) for x in s]

answer = 'ДА'

a = np.pad(lst_in, pad_width=1, mode='constant',
            constant_values=0)

# создали новую матрицу с нулевыми границами
```

#5. Вводится двумерный список размерностью 5 x 5 элементов, состоящий из целых чисел (пример ввода см. ниже). Проверьте, является ли этот двумерный список симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний. Выведите на экран ДА, если матрица симметрична и НЕТ - в противном случае.

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

Sample Input:

```
2 3 4 5 6
3 2 7 8 9
4 7 2 0 4
```

5 8 0 2 1
6 9 4 1 2

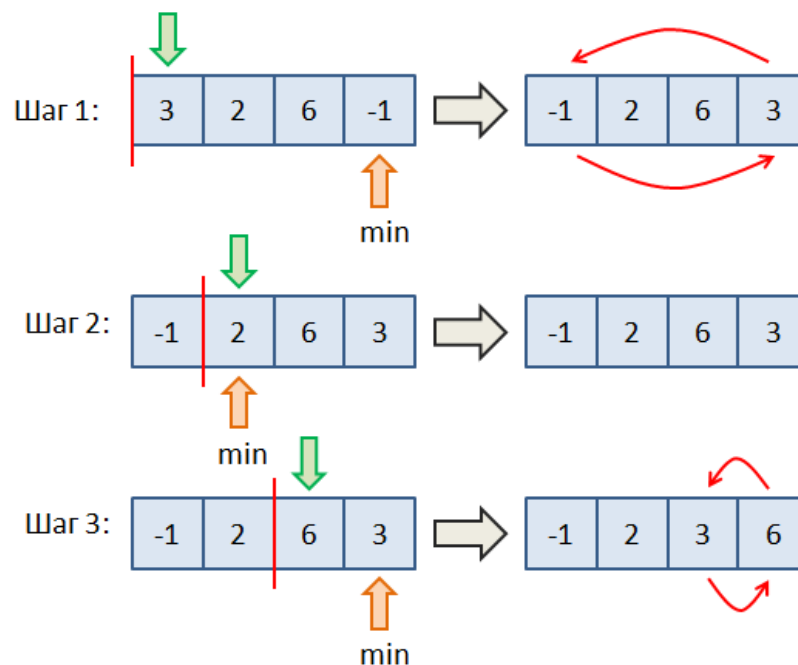
Sample Output:

ДА

```
import sys

# считывание списка из входного потока
s = sys.stdin.readlines()
lst_in = [list(map(int, x.strip().split())) for x in s]
k = 'ДА'
```

#6. Вводится список целых чисел в одну строку через пробел. Необходимо выполнить его сортировку выбором по возрастанию (неубыванию). Идея алгоритма очень проста и проиллюстрирована на рисунке ниже.



Вначале мы рассматриваем первый элемент списка и ищем второй минимальный относительно первого элемента (включая и его). На рисунке - это последний элемент со значением -1. Затем, меняем местами первый и последний элементы. Переходим ко второму элементу списка и повторяем эту же процедуру, но относительно второго элемента (то есть, первый уже не рассматриваем). На рисунке минимальный элемент - это 2, поэтому менять местами здесь ничего не нужно. Переходим к 3-му элементу со значением 6. Относительно него находим минимальный элемент - это 3. Меняем их местами.

Вот идея алгоритма сортировки выбором. Реализуйте его для вводимого списка целых чисел. Результат выведите в виде списка чисел одну строку через пробел.

Sample Input:

8 11 -53 2 10 11

Sample Output:

-53 2 8 10 11 11