Table of Contents

Add

Given two numbers print the result after adding them.

- Input format:
  Two space separated numbers

- Output format:
  Print the reuslt in one line

- Time limit: 1 Second

Input :
2 5

Output:
7

Input:
3 2

Output:
5

Max Window - Easy Version

Given an array of integers (nums), a sliding window of size k moves from the leftmost to the rightmost end of the array. Only the k numbers within the window are visible, and the window shifts right by one position with each step. Your task is to determine the maximum values within each sliding window.

- Input format:
  The first line of the input is the numbers n and k representing the length of nums and the size of the window respectively.
  The second line is nums.

- Output format:
  For the output just print one line, the maximum values within each sliding window

- Constraints:
  1 <= n <= 30
  -50 <= nums[i] <= 50
  1 <= k <= n

- Time Limit: 5 secs

Input:

```
8 3
1 3 -1 -3 5 3 6 7
```

Output:

```
3 3 5 5 6 7
```

Input:

```
2 2
9 11
```

Output:
```
11
```

Max Window - Hard Version

Given an array of integers (nums), a sliding window of size k moves from the leftmost to the rightmost end of the array. Only the k numbers within the window are visible, and the window shifts right by one position with each step. Your task is to determine and return the maximum values within each sliding window.

- Input format:
  The first line of the input is the numbers n and k representing the length of nums and the size of the window respectively.
  The second line is nums.

- Output format:
  For the output just print one line, the maximum values within each sliding window

- Constraints:
  1 <= n <= 10^5
  -10^4 <= nums[i] <= 10^4
  1 <= k <= n

- **Time Limit: 1 sec**
  Be careful about the time limit for this problem!

Input:

```
8 3
1 3 -1 -3 5 3 6 7
```

Output:

```
3 3 5 5 6 7
```

Input:

```
2 2
9 11
```

Output:

```
11
```

Maze Path Validator

There is a ball inside a maze, represented by a grid of empty spaces (0) and walls (1). The ball can roll through the empty spaces in up, down, left, or right directions. It will continue rolling until it hits a wall. Once stopped, the ball can then choose a new direction.

Your task is to write a program that, given an m x n maze, the starting position of the ball (`start = [startrow, startcol]`), and the destination (`destination = [destinationrow, destinationcol]`), determines whether the ball can come to a stop at the destination.

- Input format:

  The first line of the input contains two integers, `m` and `n`, representing the dimensions of the maze grid.
  The second line contains two space-separated integers, `startrow` and `startcol`, representing the starting position of the ball.
  The third line contains two space-separated integers, `destinationrow` and `destinationcol`, representing the destination position of the ball.
  The following `m` lines represent the maze grid, each containing `n` integers (0 or 1) separated by spaces.

- Output format:

  Print "true" or "false", indicating whether the ball can reach the destination.

- Constraints:
  - 1 <= m, n <= 100
  - 0 <= startrow, destinationrow < m
  - 0 <= startcol, destinationcol < n

- Time Limit: 3 seconds

Check the next page for input and output examples.

Input 1:

```
5 5
0 4
4 4
0 0 1 0 0
0 0 0 0 0
0 0 0 1 0
1 1 0 1 1
0 0 0 0 0
```

Output 1:

```
true
```

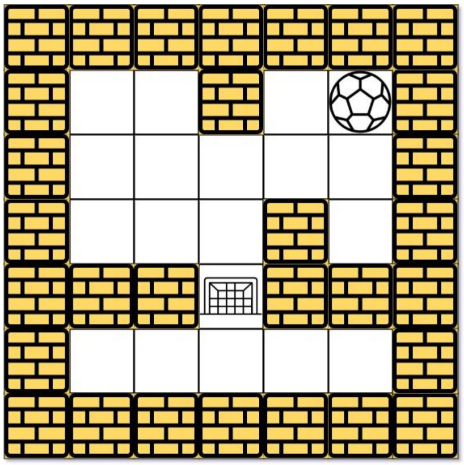
Input 2:

```
5 5
0 4
3 2
0 0 1 0 0
0 0 0 0 0
0 0 0 1 0
1 1 0 1 1
0 0 0 0 0
```

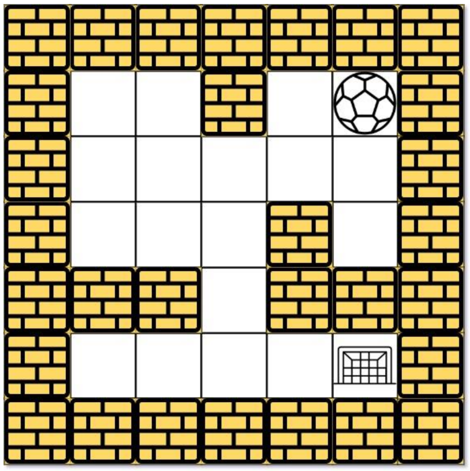
Output 2:

```
false
```

Example 1:



Example 2:



For example1: One way is: left -> down -> left -> down -> right -> down -> right

For example 2: There is no way for the ball to stop at the destination. Notice that you can pass through the destination but you can’t stop there.

First Missing Positive

Given an integer n representing the size of the array, find the smallest missing positive integer in an array. The task is to identify the first positive integer that is not present in the array.

- Input format:
  The input consists of multiple test cases so the first line of the input is the number of test cases. each test case starts with an integer n, representing the size of the array. The next line contains n integers, the elements of the array.

- Output format:
  For each test case, print the smallest missing positive integer on a separate line.

- Constraints:
  1 <= n <= 10^5
  -10^4 <= array[i] <= 10^4

- Time Limit: 2 secs

Input:

```
3
3
1 2 0
4
3 4 -1 1
5
7 8 9 11 12
```

Output:

```
3
2
1
```

Sentence Flip Adventure

In the land of words, an exciting adventure awaits - the journey to flip sentences and uncover the hidden magic of reversed words. Your task is to reverse the order of words in a sentence while keeping each word's letters unchanged.

- Input Format:

The input consists of multiple test cases. Each test case contains one or more lines, where each line is a sentence made up of words. Words are separated by spaces and consist only of letters. Each test case is separated by a star (*).

- Output Format:

For each test case, print the flipped version of the sentence, where the word order is reversed while the letters inside each word remain unchanged. Print a line with three hyphens (---) after each test case's output.

- Constraints:

The total length of sentences will not exceed 100,000 characters.
Each sentence has at most 100 words, and each word has at most 20 letters.

- Time Limit: 3 seconds

Input1:

```
Hello world
Programming is fun
Magic with words
*
```

Output1:

```
world Hello
fun is Programming
words with Magic
---
```

Input2:

```
HTML programmer?
*
```

Output2:

```
programmer? HTML
---
```

Mind Blowing Number

Given a num representing an integer, determine if it's a mind blowing number.

A mind blowing number is a number that remains the same when rotated by 180 degrees (viewed upside down).

- Input format:

A num representing the integer.

- Output format:

Print “true” if num is a mind blowing number; otherwise, print “false”.

- Time Limit: 1 second

Input1:
69

Output1:
true

Input2:
88

Output2:
true

Input3:
962

Output3:
false