***Problem Statements***

***Day 1***

***Question 1:-***

Amazon has installed WiFi routers on the houses along a straight street.

The city's buildings are arranged linearly, denoted by indices 1 to n.

There are m Amazon routers, and each has a certain range associated with it. Router j installed at a certain building location i can only provide Internet to the buildings in the range (i - routerRange[j]). (i+ routerRange[j]) inclusive, where routerRange[j] is the range parameter of router j.

A building is considered to be served if the number of routers providing internet to the building is greater than or equal to the number of people living in it. Given a list of the number of people living in each building, the locations of the buildings where the routers will be installed and each router's range, find the number of served buildings in the city.

***Test Case:-***

buildingCount = [1, 2, 1, 2, 2]

routerLocation = [3, 1]

routerRange = (1,2 )

***Question 2:-***

There is a straight line of students of various heights. The students' heights are given in the form of an array, in the order they are standing in the line.

Consider the region of a student as the length of the largest subarray that includes that student's position, and in which that student's height is equal to maximum height among all students present in that subarray. Return the sum of the region of all students.

Example:-

heights = [3, 5, 6]

***Question 3:-***

Devu and his brother love each other a lot. As they are super geeks, they only like to play with arrays. They are given two arrays a and b by their father. The array a is given to Devu and b to his brother.

As Devu is really a naughty kid, he wants the minimum value of his array a should be at least as much as the maximum value of his brother's array b.

Now you have to help Devu in achieving this condition. You can perform multiple operations on the arrays. In a single operation, you are allowed to decrease or increase any element of any of the arrays by 1. Note that you are allowed to apply the operation on any index of the array multiple times.

You need to find minimum number of operations required to satisfy Devu's condition so that the brothers can play peacefully without fighting.

Input

The first line contains two space-separated integers n, m (1 ≤ n, m ≤ 105). The second line will contain n space-separated integers representing content of the array a (1 ≤ ai ≤ 109). The third line will contain m space-separated integers representing content of the array b (1 ≤ bi ≤ 109).

Output

You need to output a single integer representing the minimum number of operations needed to satisfy Devu's condition.

input

3 2

1 2 3

3 4

output

4

***Question 4:-***

You are given a 0-indexed array nums consisting of positive integers.

There are two types of operations that you can apply on the array any number of times:

Choose two elements with equal values and delete them from the array.

Choose three elements with equal values and delete them from the array.

Return the minimum number of operations required to make the array empty, or -1 if it is not possible.

Input: nums = [2,3,3,2,2,4,2,3,4]

Output: 4

Explanation: We can apply the following operations to make the array empty:

- Apply the first operation on the elements at indices 0 and 3. The resulting array is nums = [3,3,2,4,2,3,4].

- Apply the first operation on the elements at indices 2 and 4. The resulting array is nums = [3,3,4,3,4].

- Apply the second operation on the elements at indices 0, 1, and 3. The resulting array is nums = [4,4].

- Apply the first operation on the elements at indices 0 and 1. The resulting array is nums = [].

It can be shown that we cannot make the array empty in less than 4 operations.

***Question 5:-***

There is a one-dimensional garden on the x-axis. The garden starts at the point 0 and ends at the point n. (i.e., the length of the garden is n).

There are n + 1 taps located at points [0, 1, ..., n] in the garden.

Given an integer n and an integer array ranges of length n + 1 where ranges[i] (0-indexed) means the i-th tap can water the area [i - ranges[i], i + ranges[i]] if it was open.

Return the minimum number of taps that should be open to water the whole garden, If the garden cannot be watered return -1.

Input: n = 5, ranges = [3,4,1,1,0,0]

Output: 1