

CS213M: Assignment 1

General Instructions and Tips:

- For each problem, test cases (p<x>_t<y>.txt) will be provided in the test folder. Note that your code will be tested on hidden test cases on submission.
- Use the following command to compile p1.cpp: `g++ p1.cpp`. An executable a.out is created. To run the executable, use the command `./a.out`
- USE 'cin' and 'cout' for taking input and printing respectively. To take input from a file abc.txt, use `./a.out < abc.txt`
- Try to solve the problems with any solution you think of, and try to improve the solution to make it as efficient as possible. As we will be testing on huge test cases, try to make 'intelligent' solutions. Timing your code, as explained in class, might help.
- Students are expected to adhere to the highest standards of integrity and academic honesty. Acts such as copying in the examinations and sharing code for the programming assignments will be dealt with strictly, in accordance with the institute's [procedures](#) and [disciplinary actions](#) for academic malpractice.

Definition(For P1 and P2): A *majority element* in an array A of size n is an element that appears **strictly more** than $n/2$ times (and hence there is at most one such element).

P1: Majority element in a Sorted array

Write a C++ program which checks whether an input element x is the majority element in a sorted array. It should print 1 if the element x is the majority element else prints 0

Input format: The first line contains n and x; the size of the sorted array and the element to be checked respectively.

The second line of input contains n integers

Example:

Input	Output
7 3 1 2 3 3 3 3 10	1
8 4 1 4 2 4 4 4 6 6	0
5 3 1 1 1 2 2	0

P2: Majority Element

Write a C++ program which prints the majority element if it exists, else prints -1

Input format: The first line of the input contains n, the size of the array.
The second line of input contains n integers.

Example:

Input	Output
9 3 3 4 2 4 4 2 4 4	4
8 3 3 4 2 4 2 4 4	-1

P3 : How many survive from the Claws?

There are n guilty people in a line, the i -th of them holds a claw with length L_i . The bell rings and every person kills some of people in front of him. The people are allowed to kill from left to right, i.e. the first person gets to kill first and then the second and so on. Namely, the i -th person kills the j -th person if and only if $j < i$ and $j \geq i - L_i$.

You are given lengths of the claws. You need to find the total number of alive people after the bell rings.

Input format:

The first line contains one integer n ($1 \leq n \leq 10^6$) — the number of guilty people.

Second line contains n space-separated integers L_1, L_2, \dots, L_n ($0 \leq L_i \leq 10^9$), where L_i is the length of the i -th person's claw.

Output format:

Print one integer — the total number of alive people after the bell rings.

Example:

Input	Output
4 0 1 0 10	1
2 0 0	2
10 1 1 3 0 0 0 2 1 0 3	3

P4: Do you sleep during lectures?

Your friend Divya and you attend a calculus lecture. Lecture lasts n minutes. Lecturer tells a_i theorems during the i -th minute.

Divya is really interested in calculus, though it is so hard to stay awake for all the time of lecture. You are given an array t of Divya's behavior. If Divya is asleep during the i -th minute of the lecture then t_i will be equal to 0, otherwise it will be equal to 1. When Divya is awake she writes down all the theorems she is being told — a_i during the i -th minute. Otherwise she writes nothing.

You know some secret technique to keep Divya awake for k minutes straight. However you can use it **only once**. You can start using it at the beginning of any minute between 1 and $n - k + 1$. If you use it on some minute i then Divya will be awake during minutes j such that $j \in [i, i + k - 1]$ and will write down all the theorems lecturer tells.

Your task is to calculate the maximum number of theorems Divya will be able to write down if you use your technique **only once** to wake her up.

Input format:

The first line of the input contains two integer numbers n and k ($1 \leq k \leq n \leq 10^5$) — the duration of the lecture in minutes and the number of minutes you can keep Divya awake.

The second line of the input contains n integer numbers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$) — the number of theorems lecturer tells during the i -th minute.

The third line of the input contains n integer numbers t_1, t_2, \dots, t_n ($0 \leq t_i \leq 1$) — type of Divya's behavior at the i -th minute of the lecture.

Output format:

Print only one integer — the maximum number of theorems Divya will be able to write down if you use your technique **only once** to wake her up.

Example:

Input	Output
6 3 1 3 5 2 5 4 1 1 0 1 0 0	16
5 2 2 3 8 9 8 0 1 0 0 0	20
3 1 8 1 7 0 0 0	8

In the first case the better way is to use the secret technique at the beginning of the third minute.
Then the number of theorems Divya will be able to write down will be equal to 16.

(Hint: Try to relate this question with the problem discussed in class related to maximum contiguous sum)

P5: Be Patient, it's a long *queue*!

Amit wants to enter a football stadium that has N entrances. But he is the last person to enter the stadium entrance.

There already is a queue of a_i people in front of the i_{th} entrance. Each entrance allows one person from its queue to enter the stadium in one minute.

Allen uses the following strategy to enter the fan zone:

- Initially he stands in the end of the queue in front of the first entrance.
- Each minute, if he is not allowed into the fan zone during the minute (meaning he is not the first in the queue), he leaves the current queue and stands in the end of the queue of the next entrance (or the first entrance if he leaves the last entrance).

Determine the entrance through which Amit will finally enter the fan zone.

Input format:

The first line contains a single integer N the number of entrances.

The second line contains N integers a_1, a_2, \dots, a_n : the number of people in queues. These numbers do not include Allen. There are no new people entering and joining the queue.

Output format:

Print a single integer — the number of entrance that Allen will use.

Example :

Input	Output
4 2 3 2 0	3
2 10 10	1
6 5 2 6 5 7 4	6

In the first example the number of people (not including Amit) changes as follows:

$[2,3,2,0] \rightarrow [1,2,1,0] \rightarrow [0,1,0,0]$. The number in bold is the queue Amit stands in. We see that he will enter the fan zone through the third entrance.

In the second example the number of people (not including Amit) changes as follows:

$[10,10] \rightarrow [9,9] \rightarrow [8,8] \rightarrow [7,7] \rightarrow [6,6] \rightarrow [5,5] \rightarrow [4,4] \rightarrow [3,3] \rightarrow [2,2] \rightarrow [1,1] \rightarrow [0,0]$

In the third example the number of people (not including Amit) changes as follows :

$[5,2,6,5,7,4] \rightarrow [4,1,5,4,6,3] \rightarrow [3,0,4,3,5,2] \rightarrow [2,0,3,2,4,1] \rightarrow [1,0,2,1,3,0] \rightarrow [0,0,1,0,2,0]$

Submission Guidelines:

Please adhere strictly to the following submission format. If you are not able to solve all the problems, DO NOT edit the provided file in order to help evaluate your submission. Your submission should have the following directory structure:

<roll-number>_A1

|----P1

|-----p1.cpp

|----P2

|-----p2.cpp

|----P3

|-----p3.cpp

|----P4

|-----p4.cpp

|----P5

|-----p5.cpp

Zip the folder, and name the zip file *<roll_number>_A1.zip*. **Failure to follow the submission format will lead to a penalty of marks obtained.**