Please Note

1. The labe exam is for 1 hour 45 minutes.
2. Each test case carries equal marks.
3. If you pass all test cases, you will receive a bonus marks.
4. You should not write garbage programs just to pass test cases or copy.

Lab Exam 2

November 18, 2021

1. (5 points + Bonus 2 points) Increasing Decreasing Array

   You are given an array of n numbers in which the number are strictly increasing till an index, say $i(1 < i < n)$, and after that the numbers are stricly decreasing. You have to find the maximum element in this array. To this end, you have to write a function FINDMAX(A, n). This function should return the maximum element of the array.

   You should find the maximum element only using a ***variant of binary search***. That is, the running time of the function FINDMAX(A, n) should be $O(\log n)$. Any answer for which the running time of funciton FINDMAX(A, n) is $O(n)$ will receive no marks.

   Input: The first line contains single integer $n(3 \leq n \leq 10000)$— the number of elements in the array. The next line contains n numbers seperated by a space. These numbers satisfy the condition given in the statement of the problem.

   Output: The maximum element in the array.

Input	:	5 1 2 10 9 8
Output	:	10

2. (5 points + Bonus 2 points) Operation on a Stack

   You are given a stack that contains n numbers. You can do k operations on the stack as follows:

   1. You can pop an elements from the stack.
   2. You can push an *already* popped elements on to the stack.

   You need to find the sequence of push and pop of size k that maximizes the number at the top of the stack.

   Input: The first line contains single integer $n, k(1 \leq n \leq 10000)$— the number of elements on the stack and number of operations respectively. Also $k < n$. The next line contains n numbers seperated by a space. The first number is the topmost element of the stack.

   Output: The maximum possible number on the top of the stack after k possible push and pop operations.

Input	:	5 3 1 4 3 2 5
Output	:	4

   Explaination:
   Operation 1: Pop (Pop the element 1)
   Operation 2: Pop (Pop the element 4)
   Operation 3: Push element 4.

   So, after the three operations, element 4 is on the top of the stack.

3. (7 points + Bonus 3 points) Students and BreakFast

   At the IITGN Mess, n students are standing in the queue. Today, on the menu, we have either a plain dosa or a masala dosa. Each student has a preference. He/She either want to eat a sada dosa or a masala dosa. At the front end of the queue, there is a stack of dosas, a mix of sada and masala dosas. A student at the front of the queue performs the following operation:

1. If the student likes a sada dosa and the top of the stack is a sada dosa, then the student takes the sada dosa and leaves the queue.
2. If the student likes a masala dosa and the top of the stack is a masala dosa, then the student takes the masala dosa and leaves the queue.
3. Else the student leaves the front of the queue and goes to end of the queue.

This process goes on till no one in the queue is able to leave it. All these students will not be able to have the breakfast today. Your job is to find the number of such students.

Input: In the input O(zero) will be used to represent a sada dosa and 1(one) represents a masala dosa. The first line of the input contains single integer $n (1 \leq n \leq 10000)$, the number of students in the queue as well as the number of dosas on the stack. The next line contains n integers (only 1 and 0) seperated by a space. This sequence represents the stack where the first element is the top of the stack. The third line contains n numbers $a_1, a_2, \ldots, a_n$(Each a_i is 1 and 0). The first number represents the first person of the queue who like sada dosa if $a_1 = 0$ or masala dosa if $a_1 = 1$.

Output: The number of students who will not be able to have the breakfast.

```
Input    :   3
             0 1 1
             1 0 1
Output   :   0
```

Explaination: The following happens in each iteration:

1. The student at the front goes to the back. The stack is now [0 1 1] and queue is [0 1 1].
2. The student at the front takes the sada dosa. The stack is now [1 1] and queue is [1 1].
3. The student at the front takes the masala dosa. The stack is now [1] and queue is [1].
4. The student at the front takes the masala dosa. The stack is now [] and queue is [].

4. (7 points + Bonus 3 points) Nodes Under the Subtree

You are given a binary tree rooted at a node, say node number 1. For each node x in this tree, you have to print the number of nodes in the subtree rooted at x (note that the subtree also contains the node x).

Input: The first line contains single integer $n (1 \leq n \leq 10000)$— the number of nodes in the tree. The nodes in the tree are numbered from 1 to n. The root of the tree will be node 1.

The next n lines have two numbers seperated by a space. The number x y in line i (one of these n lines) denotes the left and right child of node i in the tree respectively. For example, for the i-th node, we may have the following:

1. 5 3
   This implies that the left child of node i is 5 and the right child is 3.
2. 0 3
   This implies that node i has no left child but its right child is 3.
3. 0 0
   This implies that node i has no left or right child.

Output: Let a_i denote the number of nodes in the subtree rooted at node i. Then the ouput is the sequence a_1 a_2 $\ldots$ a_n seperated by a space.

```
Input    :   3
             2 3
             0 0
             0 0
Output   :   3 1 1
```

Explaination: In this tree, the root node 1 has one left child, that is node 2, and one right child, that is node 3. Node 2 and Node 3 have no children. Thus, the number of nodes in the subtree of node 1 is 3. The number of nodes in the subtree of node 2 and node 3 is 1.