Sort

Description:
Given an array, you should output the indices of elements according their values from small to large. For example, here is an array [2, 1, 3, 4], element 1 is the smallest one, its index is 1, so the first output is 1, element 2 is the second smallest, its index is 0, so the second output is 0, and so on. The output of array [2,1,3,4] is [1,0,2,3]. If some elements have the same value, according their indices from small to large. For example, the output of array [2,2,2,2] is [0,1,2,3]. When you print the result, you should according the format according to the **Output** description.

Input:
The first line will be an integer T (1<=T<=20), which is the number of test cases. For each test case, the first line will be the size of the array N (1 <= N<= 1000). The second line contains N integers (0<=$|a_i|$<=2^30).

Output:
The indices in output should be separated by a space.

Sample Input:
1
9
5 3 66 22 9 1 77 88 99

Sample Output:
5 1 0 4 3 2 6 7 8

A Multiplication Game

Description:
9, 3, 5, 7, 2, 9, 3, 5, 7, 2 ..... is a series in which 9, 3, 5, 7, 2 are repeated for every 5 terms. Given arbitrary numbers of indices, what is the product of all the numbers corresponding to those indices which are odd?

Input:
The first line will be an integer T (1<=T<=20), which is the number of test cases.
For each test case, the first line will be an integer P (1<=P<=19), which is the total number of indices. (Why 19?)
Followed by P lines, each of the following lines will be an index (1<= i<=2^63-1). (Why 2^63-1?)
Please be mindful of the fact that 9 is the 1st term in the series.

Output:
For valid input, your program is expected to output the product. Otherwise, it should output "1".

Sample Input:
1
2
1
7

Sample Output:
27

Hint:
The "switch" statement could be helpful.

Infinite Loop Detection

Description:
An infinite loop of states is a sequence of states that never terminates.
Given an array where the index and the value represent the current state and the next state respectively, determine whether there is any "infinite loop of states" in the array. If the value is -1, the next state would be termination.

Input:
The first line will be an integer T (1<=T<=20), which is the number of test cases.
For each test case, the first line will be the size of the array P (1 <=P<= 1000).
Followed by P lines, each of following lines will be an integer value i (-1<=i< P) in the array in order.
Please be mindful of the fact that in an array, the index starts from 0.

Output:
For each test case, for valid input, your program is expected to output "true" or "false". Otherwise, it should output "Invalid Input".

Sample Input:
1
4
2
-1
1
0

Sample Output:
false

Max Summation

Description:
Given an array A with element a1, a2, …, an, you need to find the max value of the sum of elements in the range[i,j](1<=i<=j<=n).

Input:
The first line will be an integer T (1<=T<=20), which is the number of test cases.
For each test case, the first line will be an integer N (1<=N<=1000) means the length of array A.
The second line contains n integers (0<=|a_i|<=2^30), which is the array A.

Output:
For each test case, output the the max value of the sum in the range[i,j] (1<=i<=j<=n).

Sample Input:
2
3
-2 -1 -3
4
-1 2 2 -1

Sample Output:
-1
4

Range sum

Description:
X is the 13th king of the Kingdom of Summation. Every year, there will be a great competition held in this kingdom to solve hard mathematical summation problems. This year, X gives out an interesting problem.
The problem can be described as follows:
Given an array A with element a1, a2, …, an, every element within the range [0, 100]. There are three defined operators:
<1>. I i x which is to insert x in the position i in the array, every element within the range [0, 100], i is guaranteed within the range of array length.
<2>. D x which is to delete the first occurrence of x in A
<3>. S i j which asks competitor to give the summation of elements in the range [i, j]
Now give you q operations, please output the answer every time S i j is asked.

Input:
The first line will be an integer T (1<=T<=10), which is the number of test cases.
For each test case, the first line will be an integer n (1<=n<=1000) means the length of array A.
The second line contains n integers, which is the array A.
The third line will be an integer q (1<=q<=20) means the number of operations.
The next q lines are the operations.

Output:
For each test case, output the answer every time S i j is asked in a single line.

Sample Input:
1
3
1 2 3
4
I 0 4
S 0 3
D 2
S 0 2

Sample Output:
10
8

Arrange Seats

Description:
You need to arrange K students to sit in N (N >= K) available seats in a row, at seat[1], seat[2], seat[3] ... ... seat[n]. Two seats are not necessarily adjacent. For example, seat[i] - seat[i-1] is not necessarily equal to 1. Can you make the distance between the two adjacent students as far as possible? Print the largest minimum distance.
.

Input:
The first line will be an integer T (1<=T<=10), which is the number of test cases. For each test case, the first line will be two integers N (1<=N<=1000) and K(2<=K<=N). N means the number of available seats. K means the number of student need to arrange.
The second line contains N integers (0<=i<=100), which are the position of seats (0<=seat[i]<=10^9).

Output:
For each test case, output the largest minimum distance.
Notice: One test case output one line.

Sample Input:
2
7 7
1 2 3 4 5 6 7
3 2
1 4 7

Sample Output:
1
6

HINT:
Binary search

Get My T-shirts!

Description

Bob is a typical straight man, who only has a few T-shirts and will wear them routinely in a special pattern. He stores his T-shirts his closet, and every morning he will just fetch the top one to wear, because he is too lazy to fetch the ones below, and doing this will make T-shirts collapse. However, he sometimes washes his T-shirts, and sometimes he buys some new ones. For shirts he washed, it will be stored with the same order of retrieving when next day starts. For new shirts, it will just be put on the top of other clothes (assuming new clothes are already washed). Since Bob is a person with basic hygiene habit, he will wash his T-shirts that he wore today when today ends.

As his roommate, Tom is quite interested in Bob's T-shirts, for some reason that's not suitable to talk about publicly. Your task is to help Tom to find out on given days, what is the T-shirt he wears.

Input

The first line will be an integer, indicating the number of test cases.

For each test case:

The first line contains 2 integers: N, the number of days, M, the maximum of T-shirts. (1<=N<=1000000, 1<=M<=1000000)

The second line contains a number S(S<=M), the amount of T-shirts in the closet at the beginning, followed by S integers, which is the number of these T-shirts, starting from the bottom to top.

Starting from the third line, is the operations happened on each day.

For each day:

The first number P (1<=P<=1000000) shows the total operation happened in this day.

Followed by P lines, composing of one of three possible operations: (A is the number for the the T-shirt)

buy A / wash / wear

Notice: Integers of T-shirts may duplicate. "Wear" will only happen once every day. Input data is ensured Bob will always have a T-shirt when he tries to wear one or wash one.

Output

For each time you received "wear", print the integer of the T-shirt Bob wears, and the number of T-shirts remaining in the closet at that time, in a new line. If there are no T-shirts remaining in the closet (even when there are T-shirts waiting to get washed), print 0.

Sample Input

```
1
5 10
3 1 2 3
3
wash
```

wash
wear
2
buy 4
wear
3
buy 5
wash
wear
1
wear
2
wash
wear

Sample Output

1 0
4 3
4 3
4 4
5 3

HINT

In Sample input day 1, since 1 is the T-shirt he wore on day 1, it will enter the list of clothes washed at the end, and will be put back to closet as the last one.

Wikipedia:https://en.wikipedia.org/wiki/Stack_(abstract_data_type)