



Lab 3 (Week of 4 September)

Question 1 - “Club Recruitments”

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Question 2 - Current Max

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Lab 3 (Week of 4 September)

Question 1 - “Club Recruitments”

Problem Description

A new student club is organising a huge event this semester for silver jubilee celebrations. Initially, they don't have enough members. So, they started recruiting new members in groups.

Meanwhile, tasks keep coming up within the club. **Each club member can work on only one task** during the semester.

If there is no member available (isn't occupied with a task) **when a task arises**, it will remain unattended regardless of whether new individuals are recruited later on.

Given the chronological order of task arrivals and club members' recruitment, find the number of tasks that will remain unattended.

In the input, a -1 denotes the arrival of a task (which may or may not go unattended depending on whether or not there are currently any free members to take up the task). Similarly, a non-negative number denotes the number of new recruits.

Note that once an individual is recruited for the club, they'll always be part of the club. When an individual takes up a task, they'll be occupied with it for the rest of the semester.

[Link to problem on OJ](#)

Input constraints

$$1 \leq N \leq 2 \times 10^5$$

$$A_i \in [-1, 10]$$

Input format

The first line of input contains a single integer N that denotes the number of events (either task arrival or new recruitment).

The second line of input contains N space-separated integers A_1, A_2, \dots, A_N that describe the N events as explained in the description.

Output Format

Output a single integer that denotes the number of unattended tasks.

Sample input and output

Sample Input	Sample Output
3 -1 -1 -1	2
11 -1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1	8

Solution

```
#include <stdio.h>

int main() {
    int n; scanf("%d", &n);
    int pref = 0, ans = 0;
    for (int i = 0; i < n; i++) {
        int x; scanf("%d", &x);
        pref += x;
        if (pref < 0) ans++, pref = 0;
    }
    printf("%d\n", ans);
    return 0;
}
```

Question 2 - Current Max

Problem Description

You are given a sequence of N integers A_1, A_2, \dots, A_N and while looking at the sequence in order from left to right (that is, from index 1 to N), you start by crossing the first number, and you cross a subsequent number if it is strictly greater than every number you crossed before. You keep crossing until you reach the end of the sequence.

Output all the numbers you have crossed.

[Link to problem on OJ](#)

Input constraints

$$1 \leq N \leq 2 \times 10^5$$

$$0 \leq A_i \leq 10^9$$

Input format

The first line of input contains a single integer N that denotes the size of the array.

The second line of input contains N space-separated integers A_1, A_2, \dots, A_N that describe the N elements.

Output Format

On the first line, output the count of numbers that you cross out, K . Note that this is a positive number since the first number is always crossed.

On the second line, output K space-separated integers denoting the numbers crossed out in the order in which they appear in the original array.

Sample input and output

Sample Input	Sample Output
7 1 1 3 2 4 10 9	4 1 3 4 10

Solution

```
#include <stdio.h>
```

```
int main() {  
    int n; scanf("%d", &n);  
    int final[n];  
    int sz = 0;  
    for (int i = 0; i < n; i++) {  
        int x; scanf("%d", &x);  
        if (!sz || final[sz - 1] < x) final[sz++] = x;  
    }  
    printf("%d\n", sz);  
    for (int i = 0; i < sz; i++) printf("%d ", final[i]);  
    printf("\n");  
    return 0;  
}
```