

## 오름차순으로 나열된 구간

정수의 수열  $a_1, a_2, \dots, a_n$  ( $n \geq 2$ )에서 어떤 연속적인 구간  $[a_i, a_{i+1}, \dots, a_j]$  ( $1 \leq i < j \leq n$ )에서의 정수가  $a_i \leq a_{i+1} \leq \dots \leq a_j$  인 조건을 만족하면, 이 구간을 오름차순으로 나열된 구간이라고 부른다. 또한 오름차순으로 나열된 구간  $[a_i, a_{i+1}, \dots, a_j]$ 에서  $a_{i-1} > a_i$  ( $i \neq 1$ )이고  $a_j > a_{j+1}$  ( $j \neq n$ ) 인 조건을 만족하면 이 구간을 오름차순으로 나열된 최대 연속구간(maximal increasing subsequence)이라고 부른다.

정수의 수열이 주어졌을 때, 오름차순으로 나열된 최대 연속구간의 개수와 그 구간에 포함된 모든 정수의 합을 계산하는 프로그램을 작성하시오.

예를 들어, 다음과 같은 정수의 수열에서

1 2 3 3 4 3 4 5 5 6 6 7 6 5 4 5 3 6 2 8

오름차순으로 나열된 최대 연속구간은 [1 2 3 3 4], [3 4 5 5 6 6 7], [4 5], [3 6], [2 8] 의 다섯 구간이며, 이 구간에 포함된 모든 정수의 합은 77 이다. 또한 다음과 같은 수열에서

1 2 3 4 5

오름차순으로 나열된 최대 연속구간은 [1 2 3 4 5] 뿐이며, 이 구간에 포함된 모든 정수의 합은 15 이다. 그리고 다음과 같은 수열에서

5 4 3 2 1

오름차순으로 나열된 최대 연속구간은 없다.

## 입력

입력은 표준입력(standard input)을 사용한다. 입력은  $t$  개의 테스트 케이스로 주어진다. 입력의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수  $t$ 가 주어진다. 두 번째 줄부터  $t$ 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 데이터가 입력된다. 각 줄에서 첫 번째로 입력되는 정수  $n$  ( $2 \leq n \leq 100$ )은 주어진 정수의 개수를 나타낸다. 그 다음으로는 수열에 속하는 정수가 순서대로  $n$  개가 입력된다. 이 정수는 최소 0 이며 최대 1,000 이다. 각 정수들 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

## 출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는 수열에서 오름차순으로 나열된 최대 연속구간의 개수와 그 구간들에 속하는 모든 정수를 더한 값을 출력한다. 만약 최대 연속구간이 없는 경우에는 구간의 개수를 0으로 출력하고, 모든 정수를 더한 값도 0으로 출력한다.

## 입력과 출력의 예

입력	출력
3	5 77
20 1 2 3 3 4 3 4 5 5 6 6 7 6 5 4 5 3 6 2	1 15
8	0 0
5 1 2 3 4 5	
5 5 4 3 2 1	

## 샘플 코드

```
#include <iostream>
using namespace std;

const int MAX_SIZE = 100;

int main()
{
    int numTestCases;
    int data[MAX_SIZE], size;

    cin >> numTestCases;
    for (int i = 0; i < numTestCases; i++)
    {
        cin >> size;
        for (int j = 0; j < size; j++)
            cin >> data[j];

        solve(data, size);
    }
    return 0;
}

void solve(int data[], int n)
{
}
```