Understanding recursion is not easy but unfortunately to solve this problem you need to understand it quite well. Below you can see a program written in plain C, which takes as input up to 40000, 32-bit integers and produces an output. It continues to do so until a number set of zero elements appear. Given the input your job is to find out what output will the following program will produce.

```
#include<stdio.h>
#include<math.h>
int const MAX=40000;
long nums[MAX];
long recur(int i,int j,int N)
{
    long t1=0,t2=0,t=0;
    if(i<0 || j<0 || i>=N || j>=N) return 0;
    if(i==j) t=recur(i+1,j+1,N);
    if(i<=j) t1=(nums[i]>nums[j])+recur(i,j+1,N);
    if(i>=j) t2=(nums[i]>nums[j])+recur(i,j-1,N);
    return t1+t2+t;
}
int main(void)
{
    long int i,j,N,kase=0;
    freopen("d.in","r",stdin);
    while(1)
    {
        scanf("%d",&N);
        for(i=0;i<N;i++)
            scanf("%ld",&nums[i]);
        if(N==0) break;
        printf("Case %d: %ld\n",++kase,recur(0,0,N));
    }
    return 0;
}
```

## Input

The input file contains maximum 10 sets of input. The description of each set is given below:

The first line of each set is an integer $N$ ($0 \leq N \leq 40000$) which indicates how many numbers are in this set. Each of the next $N$ lines contains a number. All these numbers are less than 2000000001.

Input is terminated by a set where the value of $N$ is zero.

## Output

For the input file produce the output that the program above will produce (assuming that it will run smoothly in the computer and no stack overflow will occur) for the given input file.

## Sample Input

```
4
1
2
3
4
2
6
1
0
```

## Sample Output

```
Case 1: 6
Case 2: 1
```