Vector of mountains **X41120_en**

Control 3, GRAU-PRO1, FIB (2014-11-26)

We describe the vertical profile of a hike in the mountains by the height (compared to sea level) for different consecutive points of the journey. One way to store this information, thus representing a mountainous profile, is by means of a vector keeping the heights of different points of the journey and, in particular, those of the peaks that were crossed in the hike.

For example, the vector

$$v = [0, 250, 515, 880, 430, 990, 1300, 1000, 700, 400, 0]$$

describes a hike with the climb of two mountains and the return to the starting point at sea level. The first peak of the journey is 880m high and the second 1300m.

Write a program that, given a mountainous profile represented by a vector, calculates the number of peaks and which of these peaks are higher than the last visited peak of the hike.

Your program must implement and use the following function:

```
// PRE: |v| ≥ 3
// POST: returns a vector containing the height of all the peaks in v (in the same order)
vector<int> obtain_peaks(const vector<int>& v);
```

which, given a vector v representing a mountainous profile, returns a vector w (with $|w| < |v|$) with all the peaks in v in the same order in which they appear in v. The returned vector must contain only the peaks of v.

Input

The input is formed by a natural $n \geq 3$ indicating the number of points in the mountainous profile. Following, there are the n heights registred at those points. The heights are integer numbers.

Output

Indicate the total number of mountain peaks for the profile described in the input, and their height. Then print the height of those peaks that are higher than the last peak in the described profile. The cases of a profile not having any peak, or not having peaks higher than the last one, should be noted conveniently.

Follow the format specified in the examples. Your code should follow the standards of style, and contain the appropriate comments.

Sample input 1

```
10
0 515 880 430 990 1300 1000 700 400 0
```

Sample output 1

```
2: 880 1300
-
```

Sample input 2

```
10
0 880 430 990 300 700 400 405 400 0
```

Sample output 2

```
4: 880 990 700 405
880 990 700
```

Sample input 3

```
5
0 100 200 300 400
```

Sample output 3

```
0:
-
```

Sample input 4

```
5
0 100 100 100 0
```

Sample output 4

```
0:
-
```

Sample input 5

```
3
100 814 400
```

Sample output 5

```
1: 814
-
```

Problem information

Author : Professorat de PRO1
Translator : Maria Blesa
Generation : 2014-11-24 00:00:07