

Problem I. The smallest distance between local maxima

Determine the smallest distance between two local maxima of a sequence of natural numbers ending with the number 0.
A local maximum is a number in a sequence that is larger than its neighbors.
If there are no two local maxima in the sequence, print the number 0.
The initial and final values are not considered as local maxima.

Specifications Input A sequence of integers is introduced, ending with the number 0 (the number 0 itself is not included in the sequence, but serves as a sign of its end).

Output

Print the answer to the problem.

Explanation of tests:

In the first test all twos are local maxima (they are larger than neighbors). The distance between the latter is the smallest.

There is no local maximum in the second test.

Examples of
input data

12 i=0 -> var1=12
4 i=1 -> var2=4
2 2 56 i=2 var2>var1 && var2>number

1 1
2 2
1 3
1 4
2 5
1 6
2 7
1 8
0

output
2 (7-5)

input data

1
2
3
0

output
0

print all local max

12
4
56
1
2
3
4
5
6
8
2
9
2
9
11
15
17
3
4
5
19
1
0



```
void distanceMax() {  
    int number;  
    int var1;  
    int var2;  
    int i = 0;  
    int minDistance = -1;  
    int localMaxposition1 = 0;  
    int localMaxposition2 = 0;  
  
    while(1) {  
        std::cin >> number;  
        if (number == 0) {  
            break;  
        }  
        else {  
            if (i == 0) {  
                var1 = number;  
            }  
            else if (i == 1) {  
                var2 = number;  
            }  
            else {  
                if (var2 > var1 && var2 > number) {  
                    std::cout << "Local max: " << var2 << std::endl;  
                    if (localMaxposition1 == 0) {  
                        localMaxposition1 = i; // localMaxposition1 = 2 (actual max = 56)  
                    }  
                    else if (localMaxposition2 == 0) {  
                        localMaxposition2 = i; // localMaxposition2 = 8 (actual max = 9)  
                        minDistance = localMaxposition2 - localMaxposition1; // minDistance = 8 - 2 = 6  
                        std::cout << "First min distance between local maxima is: " << minDistance << std::endl;  
                    }  
                    else { // new maxposition = 13 (actual max = 17)  
                        localMaxposition1 = localMaxposition2;  
                        localMaxposition2 = i;  
                        if (minDistance > localMaxposition2 - localMaxposition1) {  
                            minDistance = localMaxposition2 - localMaxposition1;  
                            std::cout << "Not first Min distance between local maxima is: " << minDistance << std::endl;  
                        }  
                    }  
                }  
                var1 = var2;  
                var2 = number;  
            }  
        }  
        i++;  
    }  
    if (minDistance == -1) {  
        std::cout << "Min distance does not exist." << std::endl;  
    }  
    else {  
        std::cout << "Final min distance between local maxima is: " << minDistance << std::endl;  
    }  
}
```