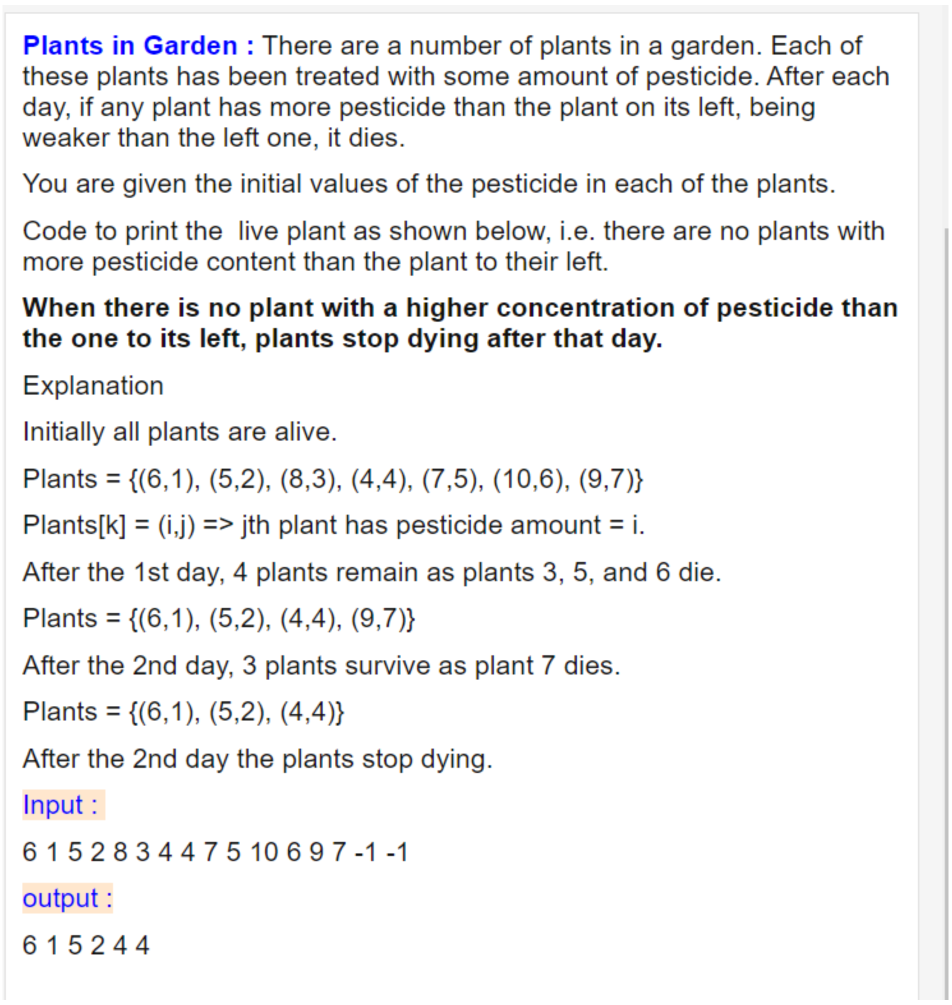


Plants in Garden : There are a number of plants in a garden. Each of these plants has been treated with some amount of pesticide. After each day, if any plant has more pesticide than the plant on its left, being weaker than the left one, it dies.

You are given the initial values of the pesticide in each of the plants.

Code to print the live plant as shown below, i.e. there are no plants with more pesticide content than the plant to their left.

When there is no plant with a higher concentration of pesticide than the one to its left, plants stop dying after that day.

Explanation

Initially all plants are alive.

Plants = {(6,1), (5,2), (8,3), (4,4), (7,5), (10,6), (9,7)}

Plants[k] = (i,j) => jth plant has pesticide amount = i.

After the 1st day, 4 plants remain as plants 3, 5, and 6 die.

Plants = {(6,1), (5,2), (4,4), (9,7)}

After the 2nd day, 3 plants survive as plant 7 dies.

Plants = {(6,1), (5,2), (4,4)}

After the 2nd day the plants stop dying.

Input :

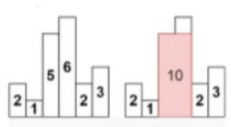
```
6 1 5 2 8 3 4 4 7 5 10 6 9 7 -1 -1
```

output :

```
6 1 5 2 4 4
```

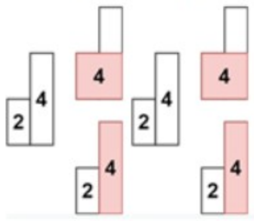
Histogram's Largest Rectangle

Given an array of integers `heights` representing the histogram's bar height where the width of each bar is `1`, return *the area of the largest rectangle in the histogram.* Submit by 29-8-2021 10.00 AM only



Input: heights = [2,1,5,6,2,3]
Output: 10
Explanation: The above is a histogram where width of each bar is 1.

The largest rectangle is shown in the red area, which has an area = 10 units.



Input: heights = [2,4]

Output: 4

Test case : (input is terminated by -1)

Input1

2 1 5 6 2 3 -1

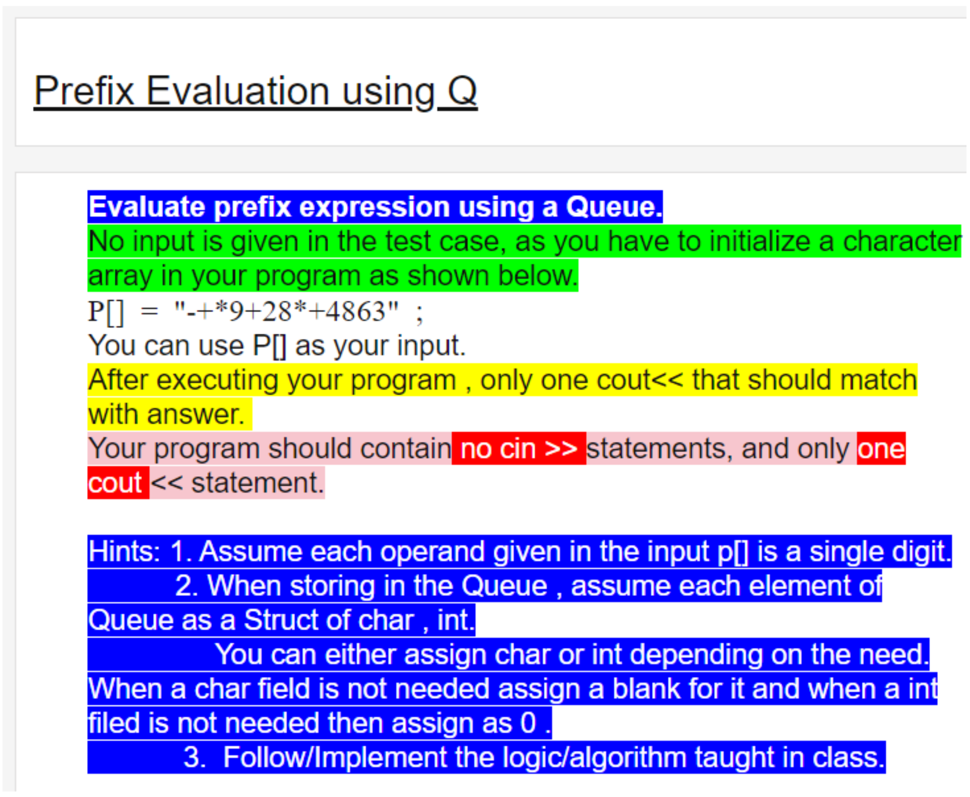
Output1

10

Input2:
2 4 -1

Output2

4



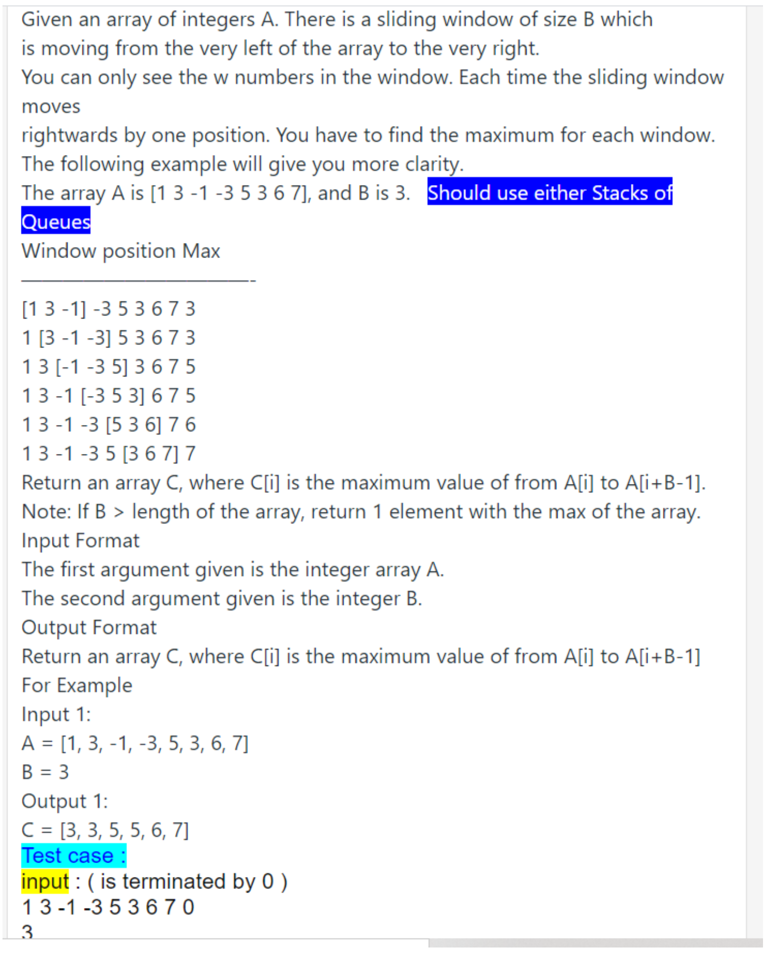
Prefix Evaluation using Q

Evaluate prefix expression using a Queue.
No input is given in the test case, as you have to initialize a character array in your program as shown below.

```
P[] = "-+*9+28*+4863" ;
```

You can use P[] as your input.
After executing your program , only one cout<< that should match with answer.
Your program should contain no cin >> statements, and only one cout << statement.

Hints: 1. Assume each operand given in the input p[] is a single digit.
2. When storing in the Queue , assume each element of Queue as a Struct of char , int.
You can either assign char or int depending on the need. When a char field is not needed assign a blank for it and when a int filed is not needed then assign as 0 .
3. Follow/Implement the logic/algorithm taught in class.



Given an array of integers A. There is a sliding window of size B which
is moving from the very left of the array to the very right.
You can only see the w numbers in the window. Each time the sliding window moves
rightwards by one position. You have to find the maximum for each window.
The following example will give you more clarity.
The array A is [1 3 -1 -3 5 3 6 7], and B is 3. Should use either Stacks of Queues
Window position Max

```
-----------------------------

[1 3 -1] -3 5 3 6 7 3
1 [3 -1 -3] 5 3 6 7 3
1 3 [-1 -3 5] 3 6 7 5
1 3 -1 [-3 5 3] 6 7 5
1 3 -1 -3 [5 3 6] 7 6
1 3 -1 -3 5 [3 6 7] 7
```

Return an array C, where C[i] is the maximum value of from A[i] to A[i+B-1].
Note: If B > length of the array, return 1 element with the max of the array.
Input Format
The first argument given is the integer array A.
The second argument given is the integer B.
Output Format
Return an array C, where C[i] is the maximum value of from A[i] to A[i+B-1]
For Example
Input 1:
A = [1, 3, -1, -3, 5, 3, 6, 7]
B = 3
Output 1:
C = [3, 3, 5, 5, 6, 7]
Test case :
input : (is terminated by 0)

```
1 3 -1 -3 5 3 6 7 0
3
```