

<https://course.acciojob.com/idle?question=d929410d-8057-4937-9384-8f6c8c634fea>

EASY

Max Score: 30 Points

First Negative Integer In Every Window Of Size K

You are given a 0-indexed array of integers `arr` of length `n` and a positive integer `k`, find the first negative integer for each and every window(contiguous subarray) of size `k`.

Note: Your task is to complete the function `printFirstNegativeInteger()` which takes the array `arr`, its size `n` and an integer `k` as inputs and returns the first negative number in every window of size `k` starting from the first till the end. If a window does not contain a negative integer , then return `0` for that window.

Input Format

The first line contains an integer `n` denoting the size of the array `arr`. The next line contains `n` space separated integers forming the array `arr`. The last line contains the window size `k`.

Output Format

Print the first negative integer in each window of size `k` seperated by a space.

Example 1

Input

9

```
-10 20 -30 -40 50 60 -70 80 90
```

```
3
```

Output

```
-10 -30 -30 -40 -70 -70 -70
```

Explanation

The windows of size 3 with their first negative integers are as follows:-

```
[-10, 20, -30] => -10
```

```
[20, -30, -40] => -30
```

```
[-30, -40, 50] => -30
```

```
[-40, 50, 60] => -40
```

```
[50, 60, -70] => -70
```

```
[60, -70, 80] => -70
```

```
[-70, 80, 90] => -70
```

Example 2

Input

```
5
```

```
-1 2 3 -4 5
```

```
2
```

Output

```
-1 0 -4 -4
```

Explanation

The windows of size 2 with their first negative integers are as follows:-

```
[-1, 2] => -1
```

```
[2, 3] => 0 (No negative element in window)
```

```
[3, -4] => -4
```

```
[-4, 5] => -4
```

Constraints

$1 \leq n \leq 10^5$

$-10^5 \leq \text{arr}[i] \leq 10^5$

$1 \leq k \leq n$

Topic Tags

[Queues](#)

[2-Pointers](#)

[Sliding Window](#)

My code

```

// in java
import java.util.*;
class Solution{
    public static int[] printFirstNegativeInteger(int arr[], int n, int k)
    {
        //Write your code here
        int firstNegativeIndex = 0;
        int firstNegativeElementofwindow;
        int ans[]=new int[n-k+1];
        int t=0;
        for(int i = k - 1; i < n; i++)
        {

            // Skip out of window and positive elements
            while ((firstNegativeIndex < i ) && (firstNegativeIndex <= i - k
|| arr[firstNegativeIndex] >= 0))
            {
                firstNegativeIndex ++;
            }

            // Check if a negative element is
            // found, otherwise use 0
            if (arr[firstNegativeIndex] < 0)
            {
                firstNegativeElementofwindow = arr[firstNegativeIndex];
            }
            else
            {
                firstNegativeElementofwindow = 0;
            }
        }
    }
}

```

```

        // System.out.print(firstNegativeElementofwindow + " ");
        ans[t++] = firstNegativeElementofwindow;
    }
    return ans;
}
}

public class Main {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for(int i = 0 ; i < n ; ++i){
            arr[i] = sc.nextInt();
        }
        int k = sc.nextInt();
        int []ans = (Solution.printFirstNegativeInteger(arr, n , k ));
        for(int i = 0 ; i < ans.length ; ++i){
            System.out.print(ans[i] + " ");
        }
    }
}

```