**Maximum of all subarrays of size k**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)[Flipkart](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Flipkart&isCmp=1)

Given an array and an integer k, find the maximum for each and every contiguous subarray of size k.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. The description of T test cases follows.  
The first line of each test case contains a single integer 'N' denoting the size of array and the size of subarray 'k'.  
The second line contains 'N' space-separated integers A1, A2, ..., AN denoting the elements of the array.

**Output:**  
Print the maximum for every subarray of size k.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N ≤ 100  
1 ≤ k ≤ N  
0 ≤A[i]<1000

**Example:**

**Input:**  
2  
9 3  
1 2 3 1 4 5 2 3 6  
10 4  
8 5 10 7 9 4 15 12 90 13

**Output:**  
3 3 4 5 5 5 6  
10 10 10 15 15 90 90

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=451>

#include<iostream>

#include<stdio.h>

#include <vector>

using namespace std;

int main(){

 int t;

 scanf("%d",&t);

 while(t--){

     int n,k;

     scanf("%d %d", &n, &k);

     int arr[n];

     for(int i =0; i<n; i++)

        scanf("%d", &arr[i]);

     std::vector<int> ans;

     for(int i =0; i<n-k+1; i++) {

         int max=-1;

         for(int j = i; j < i+k; j++) {

             max = std::max(max, arr[j]);

         }

         ans.push\_back(max);

     }

     for(int i =0; i < ans.size(); i++) {

          printf("%d ", ans[i]);

     }

     printf("**\n**");

 }

    system("pause");

 return 0;

}