Deque-STL

Problem Statement

Double ended queue or Deque(part of C++ STL) are sequence containers with dynamic sizes that can be expanded or contracted on both ends (either its front or its back). The member functions of deque that are mainly used are:

- *Deque Template:*

  ```
  std::deque<value_type>
  ```

- *Declaration:*

  ```
  deque<int> mydeque; //Creates a double ended queue of deque of int type
  ```

- *Size*

  ```
  int length = mydeque.size(); //Gives the size of the deque
  ```

- *Push*

  ```
  mydeque.push_back(1); //Pushes element at the end
  mydeque.push_front(2); //Pushes element at the beginning
  ```

- *Pop*

  ```
  mydeque.pop_back(); //Pops element from the end
  mydeque.pop_front(); //Pops element from the beginning
  ```

- *Empty*

  ```
  mydeque.empty() //Returns a boolean value which tells whether the deque is empty or not
  ```

To know more about deque, click here

Given a set of arrays of size N and an integer K, you have to find the maximum integer for each and every contiguous subarray of size K for each of the given arrays.

Input Format

First line of input will contain the number of test cases T. For each test case, you will be given the size of array N and the size of subarray to be used K. This will be followed by the elements of the array A_i.

Constraints
$1 <= T <= 1000$
$1 <= N <= 10000$
$1 <= K <= N$
$1 <= A_i <= 10000$, where A_i is the ith element in the array A.

Output Format

For each of the contiguous subarrays of size K of each array, you have to print the maximum integer.

Sample Input

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

Sample Output

```
4 6 6 4
8 8 8 10
```

Explanation

For the first case, the contiguous subarrays of size 2 are {3,4},{4,6},{6,3} and {3,4}. The 4 maximum elements of subarray of size 2 are: 4 6 6 4.

For the second case,the contiguous subarrays of size 4 are {3,4,5,8},{4,5,8,1},{5,8,1,4} and {8,1,4,10}. The 4 maximum element of subarray of size 4 are: 8 8 8 10.