Given an array & an integer k, find the maximum element for each and every contiguous sub array of size

```
#include <stdio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct Node *left;
    struct Node *right;
};

struct Node * head = NULL;
struct Node * tail = NULL;

int listIsEmpty()
{
    if(head==NULL) return 1;
    else return 0;

}

struct Node * createNode(int data)
{
    struct Node* temp =(struct Node *) malloc(sizeof(struct Node*));
    temp->data = data;
    temp->left = NULL;
    temp->right = NULL;
    return temp;
}

void addFirst(int data)
{
    struct Node *temp = createNode(data);
    if(head == NULL)
      head = tail = temp;
    else
    {
        temp->right = head;
        head->left = temp;
        head = temp;
    }
```

```
}

void printList()
{
    struct Node * temp = head;
    while(temp)
    {
        printf("%d",temp->data);
        temp = temp->right;
    }
}

void addLast(int data)
{
  struct Node * temp = createNode(data);
  if(head == NULL)
    head = tail = temp;
  else
  {
      tail->right = temp;
      temp->left = tail;
      tail = temp;
  }
}

int removeFirst()
{
    int temp = head->data;
    if(head==tail)
    {
        free(head);
        head = tail = NULL;
    }
    else
    {
       head = head->right;
       free (head->left);
       head->left = NULL;
    }
    return temp;
}

int removeLast()
{
    int temp = tail->data;
    if(head == tail)
    {
        free(head);
        head = tail = NULL;
    }
    else
    {
        tail = tail->left;
        free(tail->right);
        tail->right = NULL;
    }
    return temp;
}

int peekFirst()
{
    return(head->data);
}

int peekLast()
{
    return(tail->data);
}

slidingWindowMax(int*arr, int n, int k)
{
    addFirst(0);
    for(int i=1;i<k;i++)
    {
        while(!listIsEmpty() && arr[i]>=arr[peekLast()])
          removeLast();
        addLast(i);
    }
    for(int i=k; i<n; i++)
    {
        //printing the maximum element of the previous window
        printf("%d ",arr[peekFirst()]);
        if(peekFirst()==(i-k))
          removeFirst();
        //removing the useless elemts before inserting the next
element in the window
        while(!listIsEmpty() && arr[i]>=arr[peekLast()])
          removeLast();
        addLast(i);
        }
        printf("%d ",arr[peekFirst()]);

}

void main()
{
        int arr[] = {15,9,4,17,18,12,6,26,27,16,1,12,14,21,35,29};
        slidingWindowMax(arr,16,4);
}
```