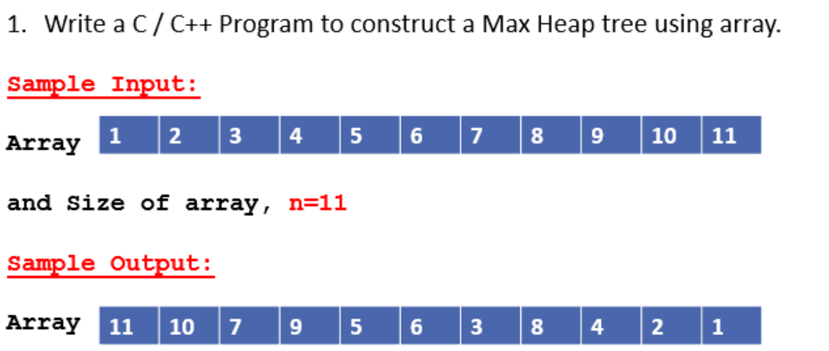
Instructions: Please read carefully

- Please rename this file as only your ID number **(e.g. 18-*****-1.doc or 18-*****-1.pdf).**
- Submit the file in the Portal Performance section labeled **Task 11. If you cannot complete the full task, do not worry. Just upload what you have completed.**

Name:- Amit Podder

ID:- 20-42273-1

Section:- [F]



1. Write a C / C++ Program to construct a Max Heap tree using array.

Sample Input:

Array

1	2	3	4	5	6	7	8	9	10	11

and Size of array, n=11

Sample Output:

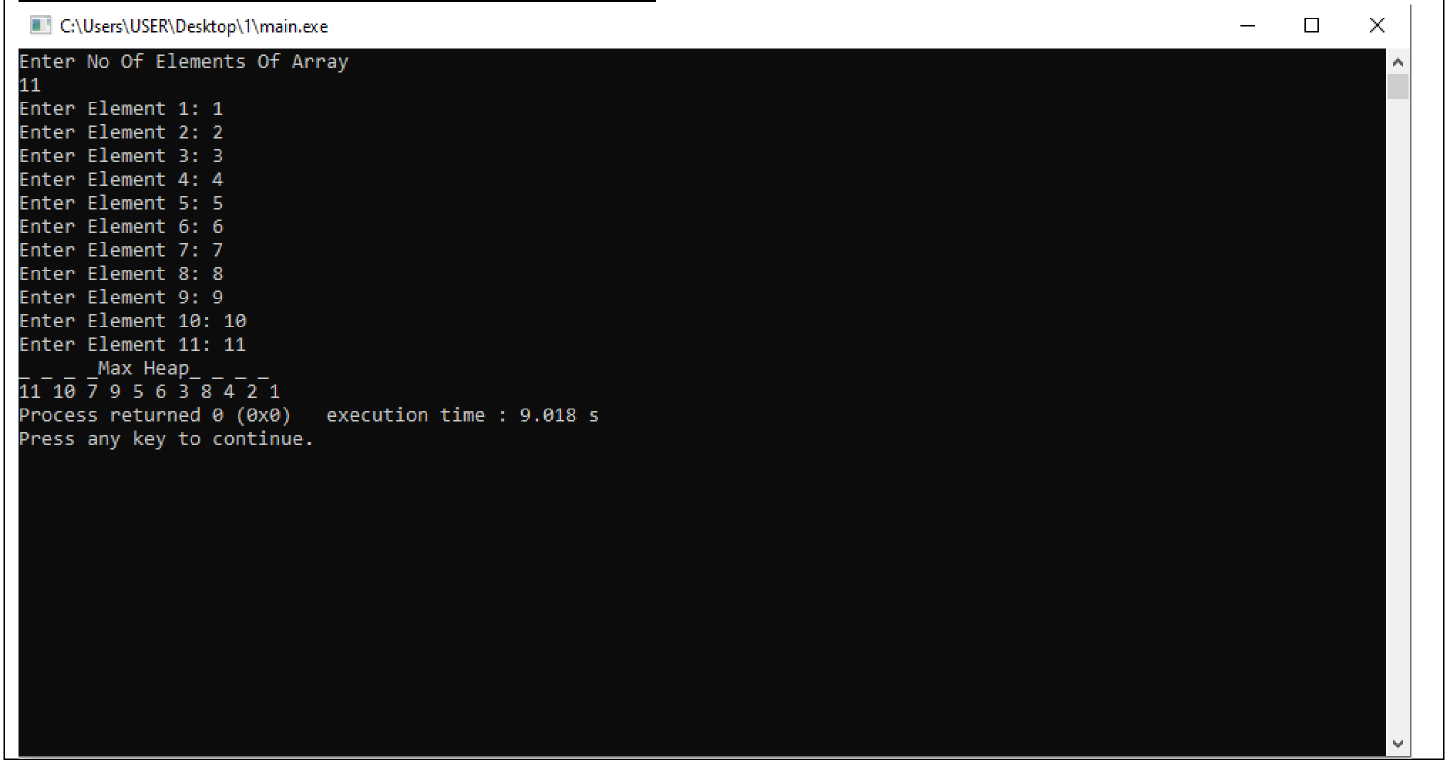
Array

11	10	7	9	5	6	3	8	4	2	1

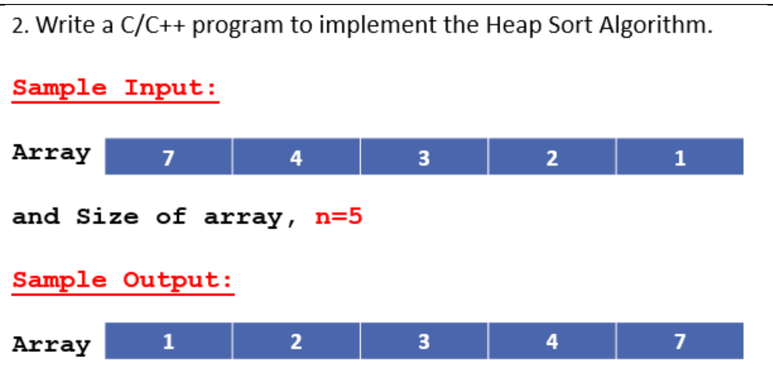
Your code here:-

```
#include <iostream>

using namespace std;

void max_heapify(int *a,int i,int n)
{
   int j,temp;
   temp=a[i];
   j =2*i;
   while(j<=n)
   {
      if(j<n && a[j+1]>a[j])
         j=j+1;
      if (temp>a[j])
         break;
      else if (temp<=a[j])
      {
         a[j/2]=a[j];
         j=2*j;
      }
   }
   a[j/2]=temp;
   return;
}

void build_maxheap(int *a,int n)
{
   int i;
   for(i=n/2;i>=1;i--)
   {
        max_heapify(a,i,n);
    }
}

int main()
{
    int n,i;
    cout<<"Enter No Of Elements Of Array"<<endl;
    cin>>n;

    int a[100];
    for (i=1;i<=n;i++)
    {
        cout<<"Enter Element "<<i<<": ";
        cin>>a[i];
    }
    build_maxheap(a,n);
    cout<<"_ _ _ _Max Heap_ _ _ _"<<endl;
    for(i=1;i<=n;i++)
    {
        cout<<a[i]<<" ";
    }
    return 0;
}
```

Your whole Screenshot here: (Console Output):-



```
Enter No Of Elements Of Array
11
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Enter Element 6: 6
Enter Element 7: 7
Enter Element 8: 8
Enter Element 9: 9
Enter Element 10: 10
Enter Element 11: 11
_ _ _ _Max Heap_ _ _ _
11 10 7 9 5 6 3 8 4 2 1
Process returned 0 (0x0)   execution time : 9.018 s
Press any key to continue.
```



2. Write a C/C++ program to implement the Heap Sort Algorithm.

Sample Input:

Array	7	4	3	2	1

and Size of array, n=5

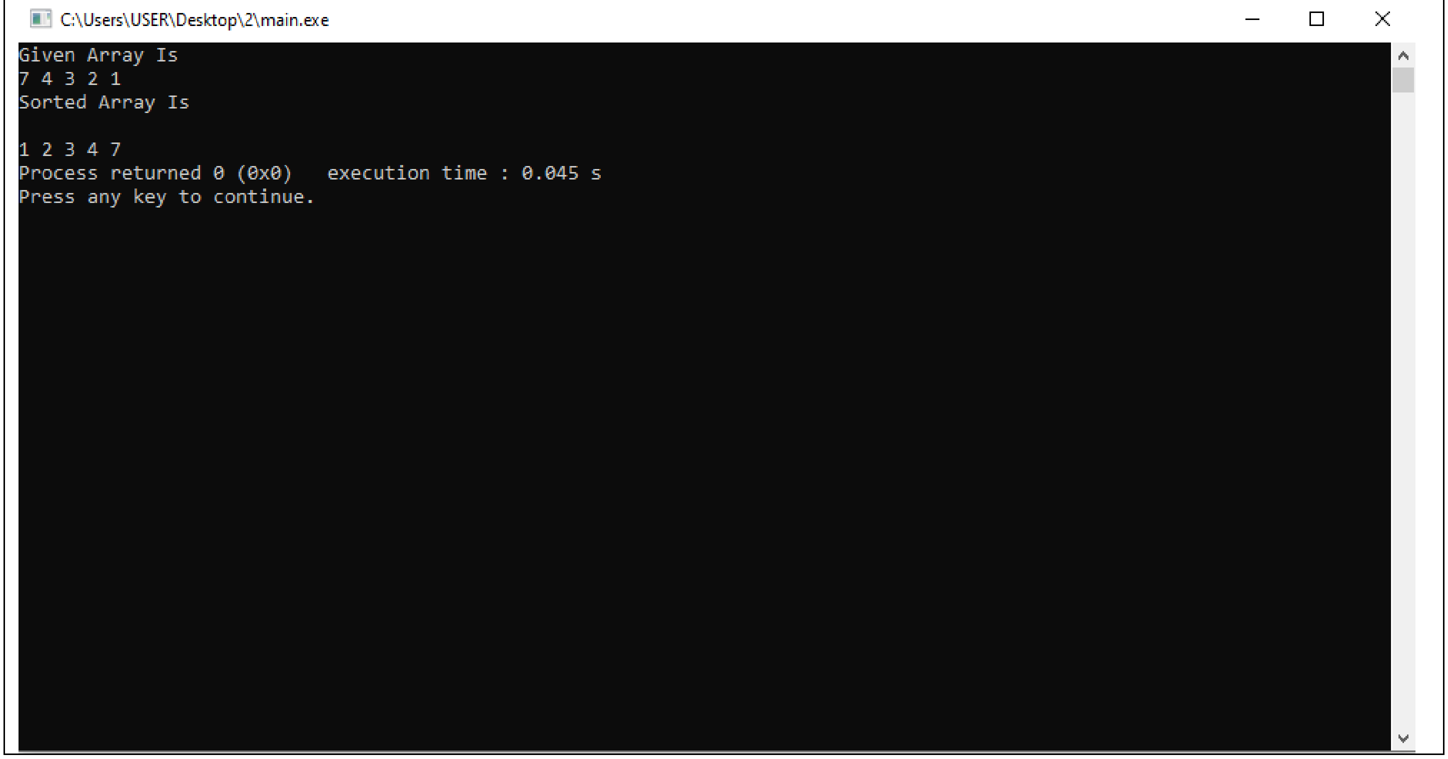
Sample Output:

Array	1	2	3	4	7

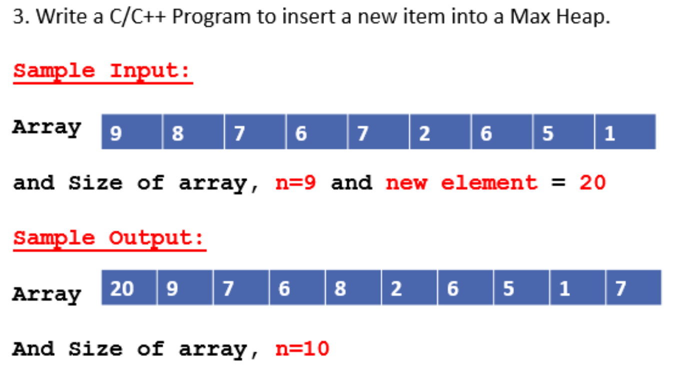
Your code here:-

```
#include<iostream>

using namespace std;

void heap(int arr[],int n,int l)
{
   int temp;
   int large=l;

   int a=2*l+1;
   int b=2*l+2;

   if(a<n && arr[a]>arr[large])
      large=a;
   if(b<n && arr[b]>arr[large])
      large=b;
   if(large !=l)
   {
      temp=arr[l];
      arr[l]=arr[large];
      arr[large]=temp;
      heap(arr,n,large);
   }

}
void heapSort(int arr[],int n)
{
   int temp;
   for(int i=n/2 - 1;i>=0;i--)
      heap(arr,n,i);
   for(int i=n-1;i>=0;i--)
   {
      temp=arr[0];
      arr[0]=arr[i];
      arr[i]=temp;
      heap(arr,i,0);
   }
}

int main()
{
   int arr[]={7,4,3,2,1};
   int n=5;

   int i;
   cout<<"Given Array Is"<<endl;

   for(i=0;i<n;i++)
   cout<<arr[i]<<" ";
   cout<<endl;

   heapSort(arr,n);
   cout<<"Sorted Array Is"<<endl;
   cout<<endl;

   for(i=0;i<n;++i)
   cout<<arr[i]<<" ";

   return 0;
}
```

<u>Your whole Screenshot here: (Console Output):-</u>



```
Given Array Is
7 4 3 2 1
Sorted Array Is

1 2 3 4 7
Process returned 0 (0x0)   execution time : 0.045 s
Press any key to continue.
```



3. Write a C/C++ Program to insert a new item into a Max Heap.

<u>Sample Input:</u>

Array	9	8	7	6	7	2	6	5	1

and Size of array, n=9 and new element = 20

<u>Sample Output:</u>

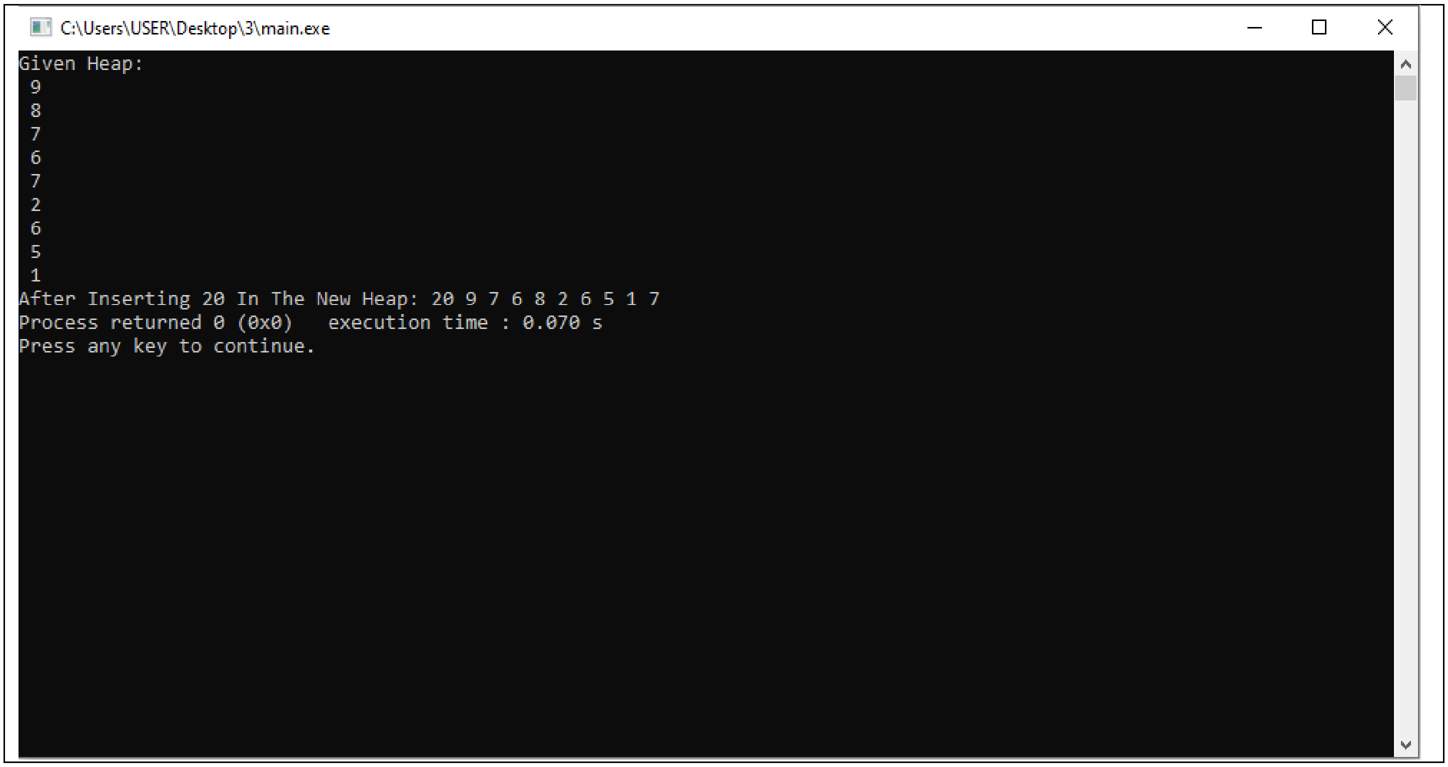
Array	20	9	7	6	8	2	6	5	1	7

And Size of array, n=10

Your code here:-

```
#include <iostream>

using namespace std;

void heap(int arr[],int n,int i)
{
   int k=(i-1)/2;
   if(arr[k]>0)
   {
      if(arr[i]>arr[k])
         {
            swap(arr[i],arr[k]);
            heap(arr,n,k);
         }
   }
}

void insertNode(int arr[],int &n,int Key)
{
   n=n+1;
   arr[n-1]=Key;
   heap(arr,n,n-1);
}

void display(int arr[],int n)
{
   for(int i=0;i<n;++i)
   cout<<arr[i]<<" ";
}

int main()
{
   int arr[]={9,8,7,6,7,2,6,5,1};
   cout<<"Given Heap: "<<endl;
   for(int x=0;x<9;x++)
   {
      cout<<" "<<arr[x]<<endl;
   }

   int n=9;
   int Key=20;
   cout<<"After Inserting 20 In The New Heap: ";
   insertNode(arr,n,Key);
   display(arr,n);

   return 0;
}
```

Your whole Screenshot here: (Console Output):-



C:\Users\USER\Desktop\3\main.exe

```
Given Heap:
 9
 8
 7
 6
 7
 2
 6
 5
 1
After Inserting 20 In The New Heap: 20 9 7 6 8 2 6 5 1 7
Process returned 0 (0x0)   execution time : 0.070 s
Press any key to continue.
```

4. Write a C/C++ Program to delete the root from a Max Heap.

Sample Input:

Array	20	15	7	6	9	2	6	5	1	7	8

and Size of array, n=11

Sample Output:

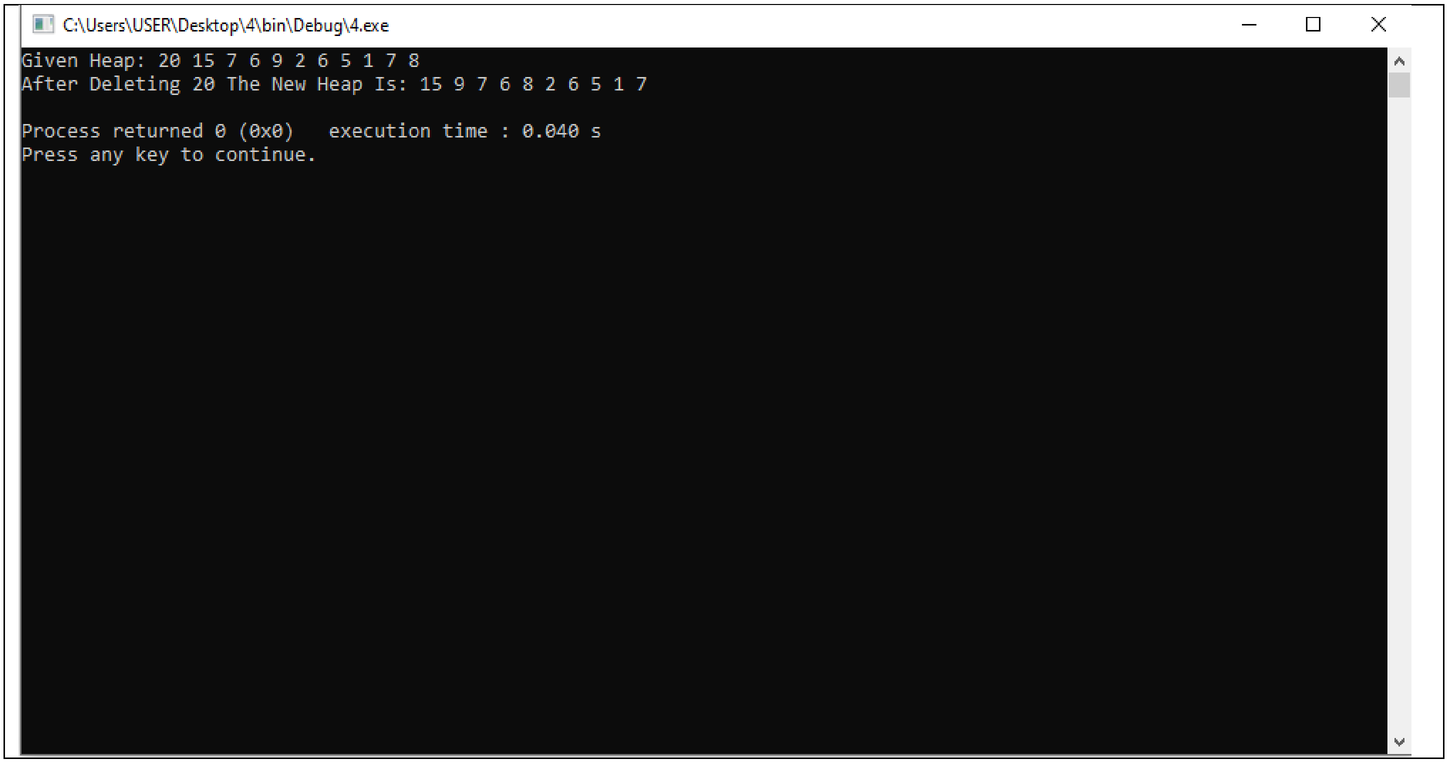
Array	15	9	7	6	8	2	6	5	1	7

and Size of array, n=10

Your code here:-

```
#include <iostream>

using namespace std;

void heap(int arr[],int n,int l)
{
   int temp;
   int large=l;

   int a=2*l+1;
   int b=2*l+2;

   if(a<n && arr[a]>arr[large])
      large=a;
   if(b<n && arr[b]>arr[large])
      large=b;
   if(large !=l)
   {
        swap(arr[l],arr[large]);
        heap(arr,n,large);
    }
}

void deletetaion(int arr[],int &n)
{
    int last=arr[n-1];
    arr[0]=last;
    n=n-1;
    heap(arr,n,0);
}

void display(int arr[],int n)
{
    for(int i=0;i<n;++i)
    cout<<arr[i]<<" ";
    cout << "\n";
}

int main()
{
    int arr[]={20,15,7,6,9,2,6,5,1,7,8};
    cout<<"Given Heap: ";

    int n=11;
    display(arr,n);
    cout<<"After Deleting 20 The New Heap Is: ";
    deletetaion(arr,n);
    display(arr,n);

    return 0;
}
```

Your whole Screenshot here: (Console Output):-



```
Given Heap: 20 15 7 6 9 2 6 5 1 7 8
After Deleting 20 The New Heap Is: 15 9 7 6 8 2 6 5 1 7

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```