//Made in alliance with Manish Jha

#include<iostream>

#include<stdlib.h>

#include <bits/stdc++.h>

using namespace std;

class HeapSort

{

public:

int \*arr;

int no\_ele;

HeapSort(int n)

{

no\_ele=n;

arr=new int[no\_ele];

}

void getdata()

{

for(int i=0;i<no\_ele;i++)

{

cout<<"Enter element : ";

cin>>arr[i];

}

}

void make\_heap()

{

for (int i = no\_ele / 2 - 1; i >= 0; i--)

min\_heap(no\_ele, i);

for (int i=no\_ele-1; i>=0; i--)

{

swap(arr[0], arr[i]);

min\_heap(i, 0);

}

cout<<"\nMin - Heap ";

showdata();

for (int i = no\_ele / 2 - 1; i >= 0; i--)

max\_heap(no\_ele, i);

for (int i=no\_ele-1; i>=0; i--)

{

swap(arr[0], arr[i]);

max\_heap(i, 0);

}

cout<<"\nMax - Heap ";

showdata();

}

void min\_heap(int n, int root)

{

int largest = root;

int l = 2\*root + 1;

int r = 2\*root + 2;

if (l < n && arr[l] > arr[largest])

largest = l;

if (r < n && arr[r] > arr[largest])

largest = r;

if (largest != root)

{

swap(arr[root], arr[largest]);

min\_heap(n, largest);

}

}

void max\_heap(int n, int root)

{

int largest = root;

int l = 2\*root + 1;

int r = 2\*root + 2;

if (l < n && arr[l] < arr[largest])

largest = l;

if (r < n && arr[r] < arr[largest])

largest = r;

if (largest != root)

{

swap(arr[root], arr[largest]);

max\_heap(n, largest);

}

}

void showdata()

{

cout<<"Sorted Data : \n-------\n";

for (int i=0; i<no\_ele; ++i)

cout << arr[i] << " ";

cout << "\n";

}

};

int main()

{

int choice,size;

cout<<"Enter the size of array : ";

cin>>size;

HeapSort s(size);

s.getdata();

do

{

cout<<endl<<endl;

cout<<"0.Exit\n1.Max Heap\n2.Min Heap\n3.Heap Sort\n";

cout<<"Enter Your Choice : "<<" ";

cin>>choice;

switch(choice)

{

case 0:

break;

case 1:

s.maxSort();

break;

case 2:

s.minSort();

break;

case 3:

s.make\_heap();

break;

default:

cout<<"invalid input"<<endl<<endl;

cout<<endl<<endl;

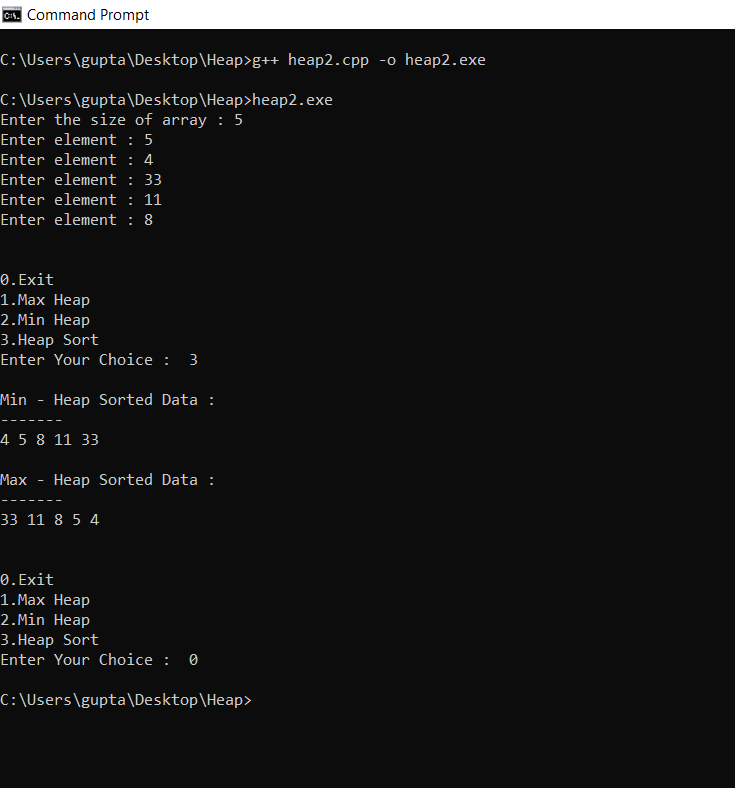
}

}while(choice!=0);

return 0;

}

**Output :**

****