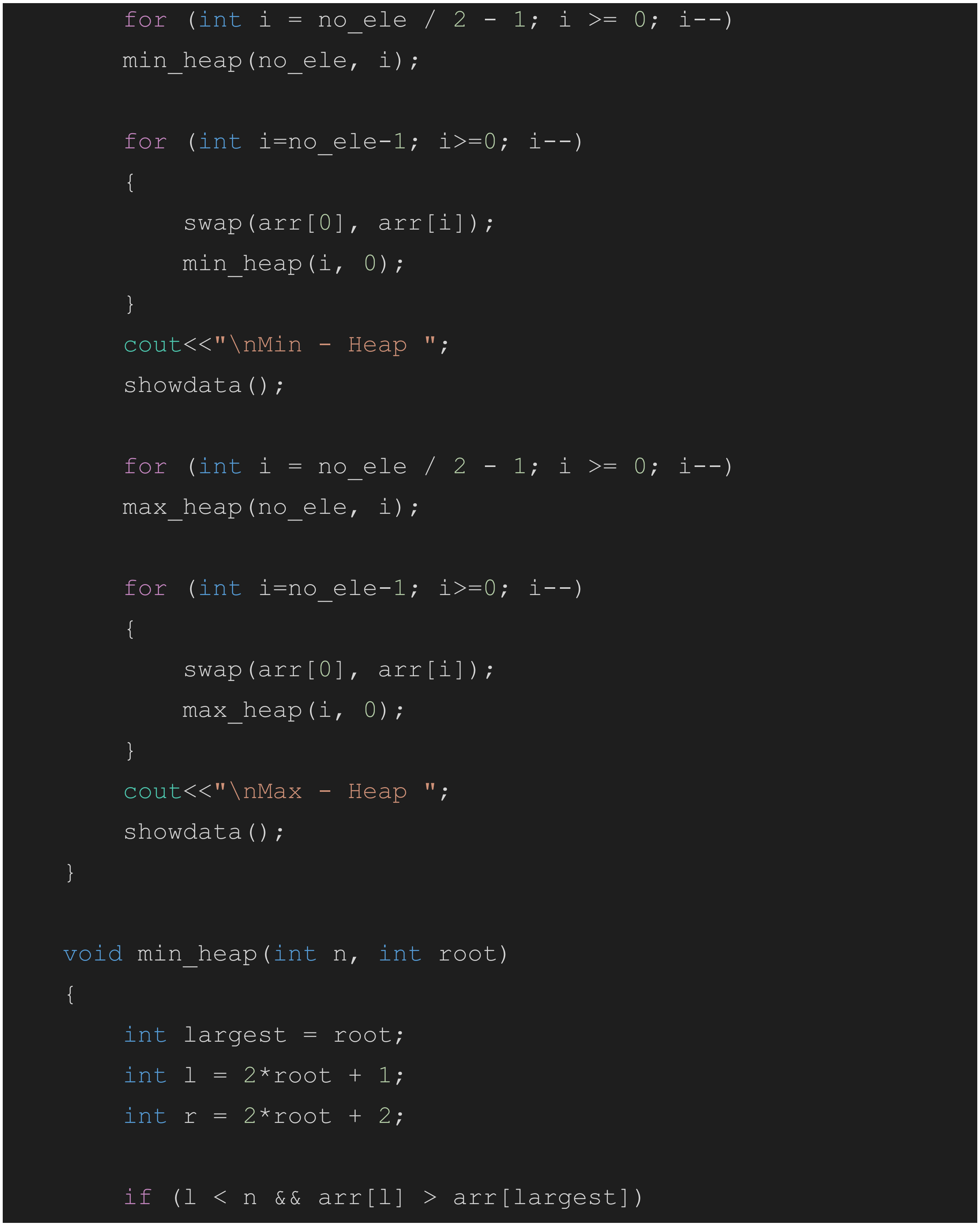
Practical Heap

Abhishek Gupta **2019450017**

```
//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);
        }
    }
```

Practical Heap

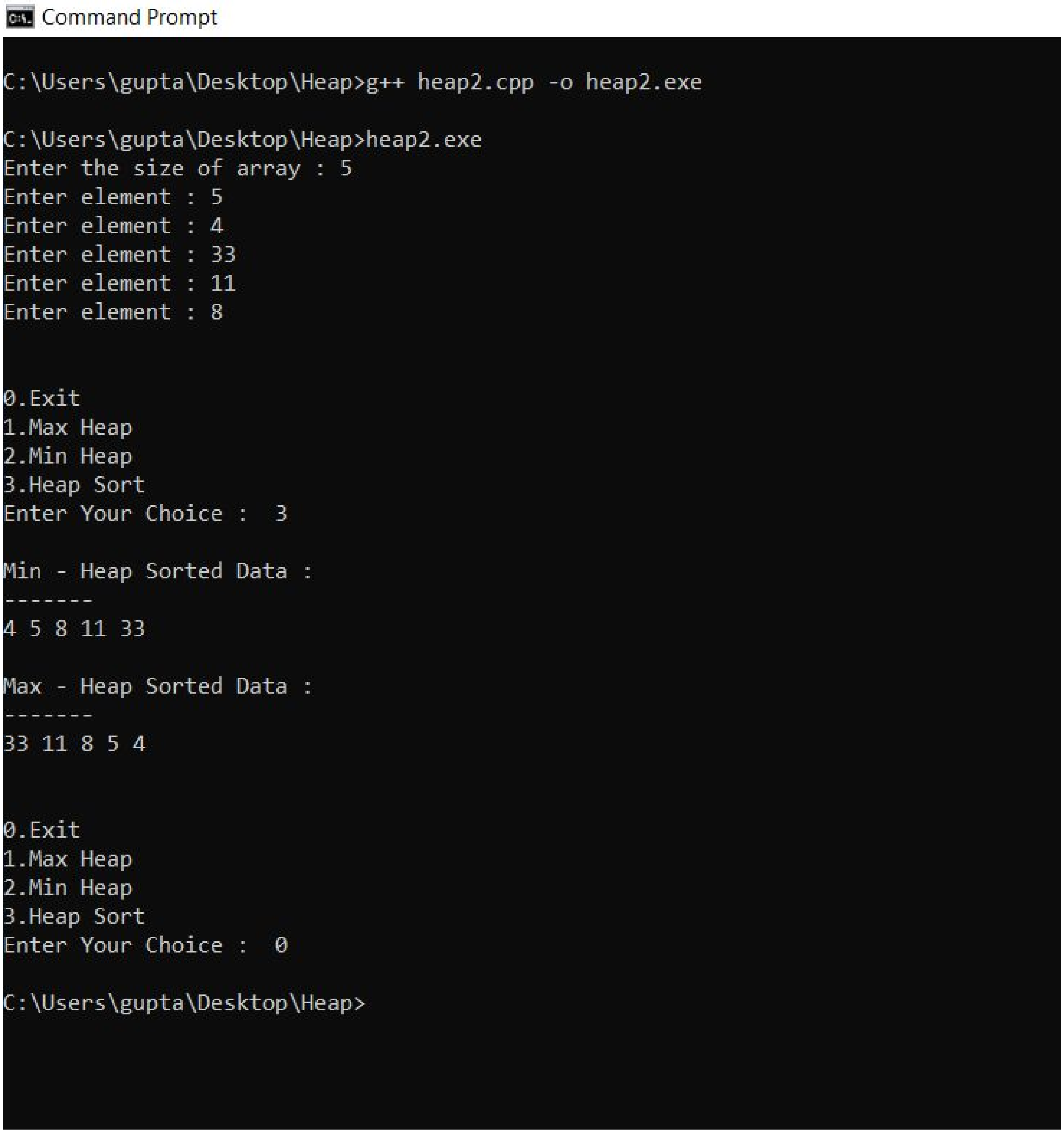
Abhishek Gupta **2019450017**

```
        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;
}
```

Practical Heap

Abhishek Gupta 2019450017

Output :



```
C:\Users\gupta\Desktop\Heap>g++ heap2.cpp -o heap2.exe

C:\Users\gupta\Desktop\Heap>heap2.exe
Enter the size of array : 5
Enter element : 5
Enter element : 4
Enter element : 33
Enter element : 11
Enter element : 8


0.Exit
1.Max Heap
2.Min Heap
3.Heap Sort
Enter Your Choice :  3

Min - Heap Sorted Data :
-------
4 5 8 11 33

Max - Heap Sorted Data :
-------
33 11 8 5 4


0.Exit
1.Max Heap
2.Min Heap
3.Heap Sort
Enter Your Choice :  0

C:\Users\gupta\Desktop\Heap>
```