EXPERIMENT NO.7

HEAP TREE:

```
#include<iostream>

using namespace std;
#define MAX 1000
class heap
{
	public:
		int no_of_elements,no_of_elements1;
		int nodes[MAX],nodes1[MAX];
	void menu(){
		for(int n=0;n<MAX;n++)
			nodes[n]=0;

		int ch=0;
		do{
			cout<<endl<<"\n1.Initialize Array \n2.Sort Ascending \n3.Sort
Descending \n4.Display \n5.Exit \n\t\t\tEnter your choice : ";
			cin>>ch;
			switch(ch){
				case 1:
					init();
					break;
				case 2:
					heap_sort_asc();
					break;
				case 3:
					heap_sort_desc();
					break;
				case 4:
					display();
					break;

				default:
					break;
			}
		}while(ch<5  && ch>=1);
	}

	void init()
	{
		cout<<"Enter number of elements";
	cin>>no_of_elements;
	int data;
	cout<<"\n\nEnter Data";
	for(int i=0;i<no_of_elements;i++)
	{
		cin>>data;
		nodes[i]=data;
	}

}

void build_tree_asc(int n)
{
	for(int i=1;i<n;i++)
	{
		if(i<=0)
			i=1;
	if(i%2==0 )
	{
				if(nodes[i]<=nodes[(i/2)-1])
					{
						int temp=nodes[i];
						nodes[i]=nodes[(i/2)-1];
						nodes[(i/2)-1]=temp;
						i=(i/2)-2;
					}
			}
	else
	{
				if(nodes[i]<=nodes[i/2])
				{
				  int temp=nodes[i];
				  nodes[i]=nodes[i/2];
				  nodes[i/2]=temp;
				  i=(i/2)-1;
				}
			}

	}
}

void build_tree_desc(int n)
{
	for(int i=1;i<n;i++)
	{
		if(i<=0)
			i=1;
```

```
        if(i%2==0 )
        {
                        if(nodes[i]>=nodes[(i/2)-1])
                        {
                         int temp=nodes[i];
                         nodes[i]=nodes[(i/2)-1];
                         nodes[(i/2)-1]=temp;
                         i=(i/2)-2;
                        }
              }
       else
       {
                         if(nodes[i]>=nodes[i/2])
                         {
                                 int temp=nodes[i];
                                 nodes[i]=nodes[i/2];
                                 nodes[i/2]=temp;
                                 i=(i/2)-1;
                            }
                  }
        }
}

void heap_sort_desc()
{
        while(no_of_elements!=0)
        {
                build_tree_desc(no_of_elements);
                int temp=nodes[0];
                nodes[0]=nodes[no_of_elements-1];
                nodes[no_of_elements-1]=temp;
                cout<<nodes[no_of_elements-1]<<"  ";
                nodes[no_of_elements-1]=0;
                no_of_elements--;
        }
}

void heap_sort_asc()
{
        while(no_of_elements!=0)
        {
                build_tree_asc(no_of_elements);
                int temp=nodes[0];
                nodes[0]=nodes[no_of_elements-1];
                nodes[no_of_elements-1]=temp;
                cout<<nodes[no_of_elements-1]<<"  ";
                nodes[no_of_elements-1]=0;
```

```
                    no_of_elements--;
               }
          }

          void display()
          {
               for(int j=0;j<=no_of_elements;j++)
               {
                    cout<<"["<<j<<"]"<<"  "<<nodes[j]<<"\n";
               }
          }
};
int main()
{
          heap h;
          h.menu();
          return 0;
}
```

OUTPUT:

```


1.Initialize Array
2.Sort Ascending
3.Sort Descending
4.Display
5.Exit
                    Enter your choice : 1
Enter number of elements6


Enter Data10 2 4 1 6 7


1.Initialize Array
2.Sort Ascending
3.Sort Descending
4.Display
5.Exit
                    Enter your choice : 2
1  2  4  6  7  10

1.Initialize Array
2.Sort Ascending
3.Sort Descending
4.Display
5.Exit
```

```
                              Enter your choice : 3


1.Initialize Array
2.Sort Ascending
3.Sort Descending
4.Display
5.Exit
                              Enter your choice : 1
Enter number of elements6


Enter Data3 2 5 7 9 10


1.Initialize Array
2.Sort Ascending
3.Sort Descending
4.Display
5.Exit
                              Enter your choice : 3
10  9  7  5  3  2
```