**Assignment No : 9**

Name : Rasika Mahure

PRN : 72018258B

Roll No : S1951093

Div : B

**Aim :-**

Implement Heap sort to sort given set of values using max or min heap.

**//Program**

#include <iostream> using namespace std;

void Maxheapify(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 { a[j/2] = a[j]; j= 2\*j;

} } a[j/2] = temp; return;

}

void BuildMaxHeap(int a[], int n){ int i; for(i=n/2; i>=2; i--){

Maxheapify(a, i, n);

}

}

void Max\_HeapSort(int a[], int n){ int i, temp; for(i=n/2; i>=2; i--){ temp=a[i]; a[i]=a[1]; a[1]=temp;

Maxheapify(a, i, n);

} } void Minheapify(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 { a[j/2] = a[j]; j= 2\*j;

}

} } void BuildMinHeap(int a[], int n){ int i; for(i=n/2; i>=2; i--){

Minheapify(a, i, n);

} } void Min\_HeapSort(int a[], int n){ int i, temp; for(i=n/2; i>=2; i--){ temp=a[i]; a[i]=a[1]; a[1]=temp;

Maxheapify(a, i, n);

} }

void Display(int a[],int n)

{

for(int i=0;i<n;i++) { cout<<a[i]<<" ";

}

}

int main() { int choice, n=10; int a[]={16,4,10,14,7,9,3,2,8,1}; do { cout<<"\n1) Build Max Heap\n"; cout<<"2) Max Heap Sort\n"; cout<<"3) Build Min Heap\n"; cout<<"4) Min Heap Sort\n"; cout<<"5) Exit\n"; cout<<"Enter Your Choice : "; cin>>choice; switch(choice)

{ case 1 :

BuildMaxHeap(a, n); Display(a,n); break; case 2 :

Max\_HeapSort(a,n); Display(a,n); break; case 3 :

BuildMinHeap(a,n); Display(a,n); break; case 4 :

Min\_HeapSort(a,n); Display(a,n); break; case 5 : break; default:

cout << "\n Invalid choice !! Please enter your choice again." << endl;

}

}while(choice!=5); return 0;

}

**// Output**

1. Build Max Heap
2. Max Heap Sort
3. Build Min Heap
4. Min Heap Sort
5. Exit

Enter Your Choice : 1

16 4 10 14 8 9 3 2 7 1

1. Build Max Heap
2. Max Heap Sort
3. Build Min Heap
4. Min Heap Sort
5. Exit

Enter Your Choice : 2

16 10 14 8 9 4 3 2 7 1

1. Build Max Heap
2. Max Heap Sort
3. Build Min Heap
4. Min Heap Sort
5. Exit

Enter Your Choice : 3

16 10 14 8 9 3 3 2 7 1

1. Build Max Heap
2. Max Heap Sort
3. Build Min Heap
4. Min Heap Sort
5. Exit

Enter Your Choice : 4

16 14 10 9 7 8 3 2 3 1 1) Build Max Heap

1. Max Heap Sort
2. Build Min Heap
3. Min Heap Sort
4. Exit

Enter Your Choice : 5