8.0. Program that implements heap tree with methods: Build, Display, Delete

#include<iostream> #include<conio.h> #include<stdlib.h> using namespace std; class heap

{

int arr[5]; public:

void insertheap(); void display(int); void reheapup(int); void buildheap(); void deleteheap();

void reheapdown(int, int);

};

void heap::buildheap()

{

for(int i=0;i<5;i++)

{

cout<<"Enter elements in heap:"; cin>>arr[i];

}

int w=1; while(w<5)

{

reheapup(w); w=w+1;

}

display(4);

}

void heap::reheapup(int i)

{

int p; if(i!=0)

{

p=(i-1)/2; if(arr[i]>arr[p])

{

int temp=arr[i]; arr[i]=arr[p]; arr[p]=temp;

}

reheapup(p);

}

}

void heap::insertheap()

{

for(int i=0;i<5;i++)

{

cout<<"Enter the element in heap:"; cin>>arr[i];

reheapup(i);

}

display(4);

}

void heap::display(int l)

{

for(int i=0;i<=l;i++)

{

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

}

}

void heap::deleteheap()

{

int l=4; arr[0]=arr[l]; l=l-1;

reheapdown(0,l);

cout<<"After deletion"; display(l);

}

void heap::reheapdown(int r,int l)

{

int lk,rk,lck,lci ; if(r\*2+1<=l )

{

lk=arr[r\*2+1]; if(r\*2+2<=l)

rk=arr[r\*2+2]; else

rk=0; //assumption if(lk>rk)

{

lck=lk; lci=r\*2+1;

}

else

{

lck=rk; lci=r\*2+2;

}

if(arr[r]<arr[lci])

{

int temp=arr[r]; arr[r]=arr[lci]; arr[lci]=temp;

}

reheapdown(lci,l);

}

}

int main()

{

int ch;

char ch1='y'; heap h;

do

{

cout<<"\n1.Insert"; cout<<"\n2.BuildHeap"; cout<<"\n3.DeleteHeap"; cout<<"\n4.Display"; cout<<"\nEnter Your choice: "; int ch;

cin>>ch; while(ch<1 || ch>6)

{

cout<<"\nPlease Enter Valid Choice"; cin>>ch;

}

switch(ch)

{

case 1:

h.insertheap(); break;

case 2:

h.buildheap(); break;

case 3:

h.deleteheap(); break;

case 4:

h.display(5); break;

default:

cout<<"Invalid choice:";

}

cout<<"\nDo you want continue?"; cin>>ch1;

}

while(ch1=='y'); return 0;

}