#include<iostream>

using namespace std;

class Heap

{

int a[50],b[50],n;

public:

Heap() { n = -1; }

void insheap(int);

int delheap();

void display();

void heapsort();

};

void Heap::insheap(int item)

{

int i, j;

if (n >= 49)

{

cout << "Heap is full!" << endl;

return;

}

n++;

i = n;

while (i > 0)

{

j = (i - 1) / 2;

if (item <= a[j])

{

a[i] = item;

return;

}

a[i] = a[j];

i = j;

}

a[0] = item;

}

int Heap::delheap()

{

if (n < 0)

{

cout << "Heap is empty!" << endl;

return -1;

}

int item = a[0];

int last = a[n];

n--;

int i = 0, left = 1, right = 2;

while (left <= n)

{

if (right <= n && a[right] > a[left])

{

left = right;

}

if (last >= a[left])

{

a[i] = last;

return item;

}

a[i] = a[left];

i = left;

left = 2 \* i + 1;

right = left + 1;

}

a[i] = last;

return item;

}

void Heap::display()

{

cout << "\nElements in heap are:\n";

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

{

cout << " " << a[i];

}

cout << endl;

}

void Heap::heapsort()

{

int item, j = 0, ch;

do

{

cout << "\nEnter item to insert: ";

cin >> item;

insheap(item);

cout << "Enter 1 to continue or 0 to stop: ";

cin >> ch;

} while (ch != 0);

cout << "\nBefore sorting\n";

display();

while (n != -1)

{

b[j] = delheap();

j++;

}

cout << "\nAfter sorting\n";

for (int i = j - 1; i >= 0; i--)

{

cout << " " << b[i];

}

cout << endl;

}

int main() {

Heap h;

h.heapsort();

return 0;

}

OUTPUT:

Enter item to insert: 34

Enter 1 to continue or 0 to stop: 1

Enter item to insert: 35

Enter 1 to continue or 0 to stop: 1

Enter item to insert: 42

Enter 1 to continue or 0 to stop: 1

Enter item to insert: 38

Enter 1 to continue or 0 to stop: 1

Enter item to insert: 43

Enter 1 to continue or 0 to stop: 0

Before sorting

Elements in heap are:

43 42 35 34 38

After sorting

34 35 38 42 43