/\*

**Name: Krushabhsingh B. Suryawanshi**

**Roll: ITSEB606**

ASSIGNMENT NO:09

AIM:-

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

#include <iostream>

using namespace **std**;

*//a function to heapify the array*

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;

*//break if the parent value is already graeter than child value*

        if (temp > a[j])

            break;

*//switching value with the parent node if temp<a[j].*

        else if (temp <= a[j])

        {

            a[j / 2] = a[j];

            j = 2 \* j;

        }

    }

    a[j / 2] = temp;

    return;

}

void **Build\_MaxHeap**(int a[], int n)

{

    int i;

    for (i = n / 2; i >= 1; i--)

    {

**MaxHeapify**(a, i, n);

    }

}

void **Max\_HeapSort**(int a[], int n)

{

    int i, temp;

    for (i = n; i >= 2; i--)

    {

*//sorting maximum value at the end*

        temp = a[i];

        a[i] = a[1];

        a[1] = temp;

*//building max heap of remaining element.*

**MaxHeapify**(a, 1, i - 1);

    }

}

void **min\_heapify**(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 if (temp >= a[j])

        {

            a[j / 2] = a[j];

            j = 2 \* j;

        }

    }

    a[j / 2] = temp;

    return;

}

void **build\_minheap**(int a[], int n)

{

    int i;

    for (i = n; i >= 2; i--)

    {

**min\_heapify**(a, i, n);

    }

}

void **Min\_HeapSort**(int a[], int n)

{

    int i, temp;

    for (i = n; i >= 2; i--)

    {

*//sorting minimum values at the end*

        temp = a[i];

        a[i] = a[1];

        a[1] = temp;

*//building max heap of remaining element.*

**min\_heapify**(a, 1, i - 1);

    }

}

void **print**(int arr[], int n)

{

    cout **<<** "\nsorted data:-";

    for (int i = 1; i <= n; i++)

        cout **<<** "->" **<<** arr[i];

    return;

}

int **main**()

{

    int n, i, ch;

    cout **<<** "\n the number of data elements to be sorted:-";

    cin **>>** n;

    int arr[n];

    for (i = 1; i <= n; i++)

    {

        cout **<<** "enter element" **<<** i **<<** ":";

        cin **>>** arr[i];

    }

*//building max heap*

    do

    {

        cout **<<** "\n1. heap sort using max heap";

        cout **<<** "\n2. heap sort using min heap";

        cout **<<** "\n3. exit";

        cout **<<** "\n enter your choice:-";

        cin **>>** ch;

        switch (ch)

        {

        case 1:

**Build\_MaxHeap**(arr, n);

**Max\_HeapSort**(arr, n);

**print**(arr, n);

            break;

        case 2:

**build\_minheap**(arr, n);

**Min\_HeapSort**(arr, n);

**print**(arr, n);

            break;

        case 3:

            return 0;

        default:

            cout **<<** "\n invalid choice" **<<** **endl**;

        }

    } while (ch != 3);

}

//:Output

Microsoft Windows [Version 6.3.9600]

(c) 2013 Microsoft Corporation. All rights reserved.

G:\it\_se>cd "g:\it\_se\Heap\_sort\" && g++ Heap\_sort.cpp -o Heap\_sort && "g:\it\_se\Heap\_sort\"Heap\_sort

the number of data elements to be sorted:-4

enter element1:11

enter element2:22

enter element3:33

enter element4:44

1. heap sort using max heap

2. heap sort using min heap

3. exit

enter your choice:-1

sorted data:-->11->22->33->44

1. heap sort using max heap

2. heap sort using min heap

3. exit

enter your choice:-2

sorted data:-->44->33->22->11

1. heap sort using max heap

2. heap sort using min heap

3. exit

enter your choice:-3

g:\it\_se\Heap\_sort>

