#include <iostream>

using namespace std;

void print(int arr[], int n);

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

void MaxHeapify(int a[], int i, int n);

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

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

void min\_heapify(int a[],int i,int n);

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

int main(){

int n, i, ch;

cout<<"\nEnter the number of data element 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<<"\n 3. exit";

cout<<"\nenter 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 !! Please enter your choice again."<<endl; } }

while(ch!=3); }

void print(int arr[], int n){

cout<<"\nSorted Data ";

for (int i = 1; i <=n; i++)

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

return; }

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

int i;

for(i = n/2; i >= 1; i--)

MaxHeapify(a, i, n); }

// 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 parent value is already greater 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 Max\_HeapSort(int a[], int n) {

int i, temp;

for (i = n; i >= 2; i--) {

// Storing 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 build\_minheap(int a[], int n) {

int i;

for(i = n/2; i >= 1; i--) {

min\_heapify(a,i,n); } }

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 Min\_HeapSort(int a[], int n) {

int i, temp;

for (i = n; i >= 2; i--) {

// Storing minimum value 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); } }

**\*OUTPUT:**

Enter the number of data element to be sorted: 5

Enter element 1: 43

Enter element 2: 23

Enter element 3: 76

Enter element 4: 345

Enter element 5: 91

1. Heap sort using max heap

2. Heap sort using min heap

3. exit

enter your choice:1

Sorted Data ->23->43->76->91->345

1. Heap sort using max heap

2. Heap sort using min heap

3. exit

enter your choice:2

Sorted Data ->345->91->76->43->23

1. Heap sort using max heap

2. Heap sort using min heap

3. exit

enter your choice:3