```
//Implement Heap sort to sort given set of values using max or min heap.
#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 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--){
temp = a[i];
a[i] = a[1];
a[1] = temp;
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/2; i>=1; i--){
min_heapify(a,i,n);
```

```
}
}
void min_HeapSort( int a[], int n){
int i, temp;
for(i=n; i>=2; i--){
temp = a[i];
a[i] = a[1];
a[1] = temp;
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<<"Enter the number of elements to be sorted: " ;
cin>>n;
int arr[n];
for(i=1; i<=n; i++) {
cout<<"Enter element "<<i<<": ";
cin>>arr[i];
}
do{
```

```
cout<<"\n\n1]Heap sort using max heap";
cout<<"\n2]Heap sort using min heap";
cout<<"\n3]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;
}
}while(ch!=3);
return 0;
}
```