```
/* Heap Sort: 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 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 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--)
	{
		// 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 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--)
        {
                // 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);
        }
}

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<<"\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);

 }
```

```
Enter the number of data element to be sorted: 8
Enter element 1: 5
Enter element 2: 7
Enter element 3: 1
Enter element 4: 3
Enter element 5: 6
Enter element 6: 4
Enter element 7: 7
Enter element 8: 10

1. Heap sort using  max heap
2. Heap sort using  min heap
 3. exit
enter your choice:1

Sorted Data ->1->3->4->5->6->7->7->10
1. Heap sort using  max heap
2. Heap sort using  min heap
 3. exit
enter your choice:2

Sorted Data ->10->7->7->6->5->4->3->1
1. Heap sort using  max heap
2. Heap sort using  min heap
 3. exit
enter your choice:3


...Program finished with exit code 0
Press ENTER to exit console.
```