```
#include <iostream>


using namespace std;


class Heap {
public:
   void heapifyA(int arr[], int n, int i);               // Function to Create a Min Heap
   void heapifyD(int arr[], int n, int i);               // Function to Create a Max Heap
   void printArray(int arr[], int n);                    // Function to print the Array
   void heapSortAscending(int arr[], int n);             // Function to perform Heap Sort in ascending
   void heapSortDescending(int arr[], int n);            // Function to perform Heap Sort in descending
   void swap(int &a, int &b);                            // Function to swap two elements
   void insert(int arr[], int n);                        // Function to Insert elements in array
};



//************* SWAP FUNCTION ************* //
void Heap::swap(int &a, int &b) {
   int temp = a;
   a = b;
   b = temp;
}



//************* PRINTING THE ARRAY ************* //
void Heap::printArray(int arr[], int n) {
   for (int i = 0; i < n; i++) {
      cout << arr[i] << " ";
   }
   cout << endl;
}
```

```
//************* MAX HEAP ************* //
void Heap::heapifyA(int arr[], int n, int i) {
   int largest = i;                                    // Root Node
   int left = 2 * i + 1;                               // Left Child
   int right = 2 * i + 2;                              // Right Child

   if (left < n && arr[largest] < arr[left]) {                 // Left child Greater than root
      largest = left;
   }

   if (right < n && arr[largest] < arr[right]) {               // Right child Greater than root
      largest = right;
   }

   if (largest != i) {
      swap(arr[largest], arr[i]);
      heapifyA(arr, n, largest);
   }
}

//************* MIN HEAP ************* //
void Heap::heapifyD(int arr[], int n, int i) {
   int smallest = i;                                   // Root Node
   int left = 2 * i + 1;                               // Left Child
   int right = 2 * i + 2;                              // Right Child

   if (left < n && arr[smallest] > arr[left]) {                // Left child Greater than root
      smallest = left;
   }

   if (right < n && arr[smallest] > arr[right]) {              // Right child Greater than root
      smallest = right;
   }
```

```
    if (smallest != i) {
        swap(arr[smallest], arr[i]);
        heapifyD(arr, n, smallest);
    }
}



//************* HEAP SORT IN ASCENDING ORDER ************* //
void Heap::heapSortAscending(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--) {                // Heapify Operation
        heapifyA(arr, n, i);
    }
    for (int i = n - 1; i > 0; i--) {
        swap(arr[0], arr[i]);                          // Swap Elements
        heapifyA(arr, i, 0);
    }
}



//************* HEAP SORT IN DESCENDING ORDER ************* //
void Heap::heapSortDescending(int arr[], int n){
    for (int i = n / 2 - 1; i >= 0; i--) {                // Heapify Operation
        heapifyD(arr, n, i);
    }
    for (int i = n - 1; i > 0; i--) {
        swap(arr[0], arr[i]);                          // Swap Elements
        heapifyD(arr, i, 0);
    }
}



//************* INSERT ELEMENT IN ARRAY ************* //
```

```
void Heap::insert(int arr[], int n) {
    cout << "Enter " << n << " elements: ";                    // Insertion in an Array
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
}



//************* MAIN FUNCTION ************* //
int main() {
    Heap h;
    int n, choice;

    cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];
    h.insert(arr, n);
    while (true) {
        cout << "1. Ascending order\n2. Descending order\n3. Exit\nEnter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout<<"\nArray: ";
                h.printArray(arr, n);
                h.heapSortAscending(arr, n);
                cout << "\nSorted Elements are: ";
                h.printArray(arr, n);
                cout<<endl;
                break;

            case 2:
```

```
            cout<<"\nArray: ";
            h.printArray(arr, n);
            h.heapSortDescending(arr, n);
            cout << "\nSorted Elements are: ";
            h.printArray(arr, n);
            cout<<endl;
            break;

          case 3:
            cout << "Exiting" << endl;
            return 0;

          default:
            cout << "Invalid choice. Please enter 1, 2, or 3." << endl;
        }
    }

    return 0;
}
```

```
Enter the number of elements: 10

Enter 10 elements: 8

5

3

11

4

7

2

17

15

12

1. Ascending order

2. Descending order

3. Exit

Enter your choice: 1


Array: 8 5 3 11 4 7 2 17 15 12


Sorted Elements are: 2 3 4 5 7 8 11 12 15 17


1. Ascending order

2. Descending order

3. Exit

Enter your choice: 2


Array: 2 3 4 5 7 8 11 12 15 17


Sorted Elements are: 17 15 12 11 8 7 5 4 3 2


1. Ascending order

2. Descending order

3. Exit

Enter your choice: 3
```

```
Exiting

Process returned 0 (0x0)   execution time : 30.947 s
Press any key to continue.
```