Implementation of Heap-sort Algorithm Using Min-Heapify and Max-Heapify:-

We have taken a global array to store the heap.
Following are the various functions used in the program:-

1. **max_heapify()**: Using this function, we are constructing a max-heap. max_heapify() is called recursively and we traverse in a bottom-up approach to the root node, also make sure the tree obeys the max-heap property.

2. **buildMaxHeap():** Using this function we are building a heap and then running the max_heapify() on the constructed heap tree.

3. **insert_max():** This function is used to insert an element into the heap tree and then max_heapify is called to maintain the max-heap property.

4. **delete_max():** This function is used to delete the max element from the tree(i.e. The root element) and then we again call max_heapify() to make sure the tree obeys the max-heap property.

5. **heapsort_max():** Using this function we sort the elements present in the max-heap tree.

6. **min_heapify():** Using this function, we are constructing a min-heap. min_heapify() is called recursively and we traverse in a bottom-up approach to the root node, also make sure the tree obeys the min-heap property

7. **buildMinHeap():** Using this function we are building a heap and then running the min_heapify() on the constructed heap tree.

8. **insert_min():** This function is used to insert an element into the heap tree and then min_heapify is called to maintain the min-heap property.

9. **delete_min():** This function is used to delete the min element from the tree(i.e. The root element) and then we again call min_heapify() to make sure the tree obeys the min-heap property.

10. **heapsort_min():** Using this function we sort the elements present in the max-heap tree.

11. **printArray():** This function is used to print the heap which is stored in an array.

12. **main():** Program execution starts from the main() function. The user can choose from the available choices and can perform the various operations(like building a heap, insertion, search, and deletion on both min-heap and max-heap).

Output:-

```
D:\Documents\DSP assignment\IT__Assignment8\maxHeap_minHeap.exe

Implementation of Heap-sort Algorithm Using Min-Heapify and Max-Heapify
--------------------------------------------------------------------------

Enter choice:
1. Using max_heapify
2. Using min_heapify
3. Exit
1

Operations on Max_heap :-
11. Build Max_heap by inserting integers
12. Insert a key into heap
13. Delete a key from heap
14. Sort the elements
15. Retun to MENU
16. Exit
11

Enter the no of elements (MAX = 100) :- 5
12 34 5 6 7

Max_heap :-
34 12 5 6 7

Operations on Max_heap :-
11. Build Max_heap by inserting integers
12. Insert a key into heap
13. Delete a key from heap
14. Sort the elements
15. Retun to MENU
16. Exit
12

Enter the element to be inserted: 18

Max_heap after insertion operation:-
34 12 18 6 7 5
```

```
D:\Documents\DSP assignment\IT__Assignment8\maxHeap_minHeap.exe

Enter the element to be inserted: 18

Max_heap after insertion operation:-
34 12 18 6 7 5

Operations on Max_heap :-
11. Build Max_heap by inserting integers
12. Insert a key into heap
13. Delete a key from heap
14. Sort the elements
15. Retun to MENU
16. Exit
13

Max_heap after deletion opeartion:-
18 12 5 6 7

Operations on Max_heap :-
11. Build Max_heap by inserting integers
12. Insert a key into heap
13. Delete a key from heap
14. Sort the elements
15. Retun to MENU
16. Exit
14

Max_heap after sorting:-
5 6 7 12 18

Operations on Max_heap :-
11. Build Max_heap by inserting integers
12. Insert a key into heap
13. Delete a key from heap
14. Sort the elements
15. Retun to MENU
16. Exit
15
```

```
D:\Documents\DSP assignment\IT__Assignment8\maxHeap_minHeap.exe
Enter choice:
1. Using max_heapify
2. Using min_heapify
3. Exit
2

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
24. Sort the elements
25. Retun to MENU
26. Exit
21

Enter the no of elements (MAX = 100) :- 5
12 34 5 6 7

Min_heap :-
5 6 12 34 7

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
24. Sort the elements
25. Retun to MENU
26. Exit
22

Enter the element to be inserted: 18

Min_heap after insertion operation:-
5 6 12 34 7 18

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
```

```
D:\Documents\DSP assignment\IT__Assignment8\maxHeap_minHeap.exe
26. Exit
22

Enter the element to be inserted: 18

Min_heap after insertion operation:-
5 6 12 34 7 18

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
24. Sort the elements
25. Retun to MENU
26. Exit
23

Min_heap after deletion operation:-
6 7 12 34 18

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
24. Sort the elements
25. Retun to MENU
26. Exit
24

Min_heap after sorting:-
34 18 12 7 6

Operations on Min_heap :
21. Build Min_heap by inserting integers
22. Insert a key into heap
23. Delete a key from heap
24. Sort the elements
25. Retun to MENU
26. Exit
```