19. Heap Sort

Aim:

To arrange a series of numbers using Heap Sort.

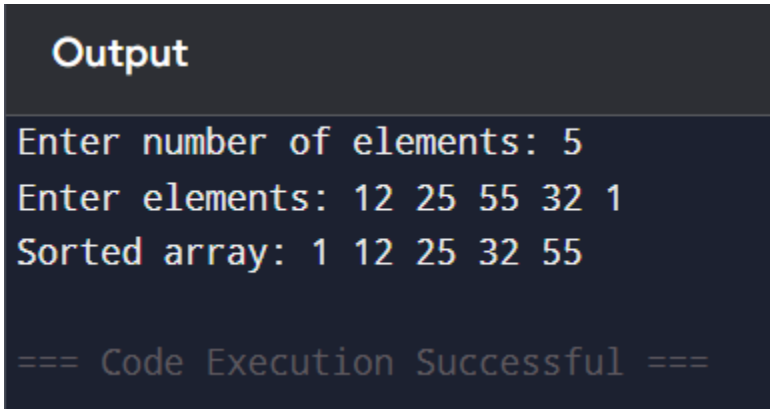
Algorithm:

1. Build a max heap from the input array.
2. Swap the root (largest element) with the last element.
3. Heapify the reduced heap.
4. Repeat until the array is sorted.

CODE:

```
#include <stdio.h>

void heapify(int arr[], int n, int i) {
    int largest = i, l = 2 * i + 1, r = 2 * i + 2;

    if (l < n && arr[l] > arr[largest]) largest = l;
    if (r < n && arr[r] > arr[largest]) largest = r;

    if (largest != i) {
        int temp = arr[i]; arr[i] = arr[largest]; arr[largest] = temp;
        heapify(arr, n, largest);
    }
}

void heapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--) heapify(arr, n, i);
    for (int i = n - 1; i >= 0; i--) {
        int temp = arr[0]; arr[0] = arr[i]; arr[i] = temp;
        heapify(arr, i, 0);
    }
}
```

```
int main() {
    int n, arr[20];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter elements: ");
    for (int i = 0; i < n; i++) scanf("%d", &arr[i]);

    heapSort(arr, n);

    printf("Sorted array: ");
    for (int i = 0; i < n; i++) printf("%d ", arr[i]);
    return 0;
}
```



RESULT:

The program successfully executed and displayed the heap sort method.