Experiment 19: Heap Sort

Aim:
To write a C program to arrange a series of numbers using Heap Sort.

Algorithm:

1. Start the program.
2. Build a max heap from input array.
3. Swap the root with the last element.
4. Reduce heap size and heapify the root.
5. Repeat until the array is sorted.
6. Stop.

Code:

```
#include <stdio.h>
void heapify(int arr[], int n, int i) {
    int largest = i;
    int l = 2 * i + 1;
    int 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 arr[6] = {12, 11, 13, 5, 6, 7}, n = 6;
    heapSort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

Sample Output:

```
Sorted array: 5 6 7 11 12 13

=== Code Execution Successful ===
```

Result:
The program successfully sorts numbers using Heap Sort.