Name :- Aditya Dilip Jadhav

Roll No :- 55

Class: SE Section :- C

Experiment 7 :- Implementation Heap Sort Algorithm.

```
#include <stdio.h>
 int i;
// Function to swap the the position of two elements
void swap(int *a, int *b)
{
int temp = *a;
*a = *b;
*b = temp;
}

void heapify(int arr[], int n, int i)
{
// Find largest among root, left child and right child
int largest = i;
int left = 2 * i + 1;
int right = 2 * i + 2;

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

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

// Swap and continue heapifying if root is not largest
 if (largest != i)
{
swap(&arr[i], &arr[largest]);
 heapify(arr, n, largest);
}
}

// Main function to do heap sort
void heapSort(int arr[], int n)
{
// Build max heap
for (i = n / 2 - 1; i >= 0; i--)
heapify(arr, n, i);

// Heap sort
for (i = n - 1; i >= 0; i--)
{
swap(&arr[0], &arr[i]);

// Heapify root element to get highest element at root again
 heapify(arr, i, 0);
}
}

// Print an array
void printArray(int arr[], int n)
{
for (i = 0; i < n; ++i)
printf("%d ", arr[i]);
printf("\n");
}
```

```
// Driver code
int main()
{
int arr[] = {1, 12, 9, 5, 6, 10};
int n = sizeof(arr) / sizeof(arr[0]);
heapSort(arr, n);
printf("Sorted array is \n");
 printArray(arr, n);
}
```

OUTPUT :-

```
Sorted array is
1 5 6 9 10 12

--------------------------------------------
Process exited after 0.02295 seconds with return value 0
Press any key to continue . . .
```