Lab 10

Prasanna Natarajan
1410110298

Code:

```
/*
Name           :   Prasanna Natarajan
Roll Number    :   1410110298
Inputs         :   An integer array
Outputs        :   Sorted array
*/

#include<stdio.h>
#include<stdlib.h>

// function defenitions
int* buildMaxHeap(int arr[],int n);
void heapSort(int arr[], int n);
void MaxHeapify(int arr[], int n, int i);
// main function
int main(){

    int arr[] = {1,2,3,4,5}; //input array
    int n = 5;  //size of array

    heapSort(arr,n); //calling heapSort
    int i;
    for(i=0;i<n;i++){
        printf("%d ",arr[i]); // printing the sorted array
    }
    printf("\n");

    return 0;
}
// function heapsort
// inputs: array to be sorted and size of array
void heapSort(int arr[], int n){

    arr = buildMaxHeap(arr,n);
    int i = n-1;
    for(i=n-1;i>=0;i--){
        int temp = arr[0];
        arr[0] = arr[i];
        arr[i] = temp;
        MaxHeapify(arr, i, 0);
    }
    int j;
    int *a = malloc(sizeof(int)*n);
    for(i=0,j=n-1;i<n,j>=0;i++,j--){
        a[j] = arr[i];
```

```
    }
    for(i=0;i<n;i++){
        arr[i] = a[i];
    }

}

// function buildMaxHeap
// inputs: integer array and size of the array
int* buildMaxHeap(int arr[],int n){
    int i;
    for (i=(n/2)-1;i>=0;i--)
        MaxHeapify(arr, n, i);

    return arr;
}

// function MaxHeapify
// inputs: integer array as heap, size of the heap and root index
void MaxHeapify(int arr[], int n, int i){

    int left = (2*i)+1;
    int right = (2*i)+2;

    int largest;

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

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


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

Screenshot: