Nisarg Wath
642403014

Assignment 6

HEADER.H

```c
typedef struct heap{
    int * data;
    int rear,size;
}heap;

void initHeap(heap * h1, int size);
void swaph(heap * h1, int a, int b);
int parent(int index);
void insertInheap(heap * h1, int data);
void print_heap(heap h1);

void heapify(heap * h1);
void delete_max(heap * h1);
void heap_sort(heap *h1);
```

LOGIC.C

```c
#include<stdio.h>
#include<stdlib.h>
#include "header.h"



void initHeap(heap * h1, int size){
    h1->data = (int *)malloc(sizeof(int) * size);
    h1->rear =-1;
    h1->size=size;
    return;
}
void swaph(heap * h1, int a, int b){
    if(a>=h1->size || b>=h1->size) return;

    int temp = h1->data[a];
    h1->data[a] =h1->data[b];
    h1->data[b] = temp;
    return;
}
int parent(int index){
    return ((index -1)/2);
}
```

```
void insertInheap(heap * h1, int data){
    if(h1->rear == h1->size-1) return;
    h1->rear++;
    h1->data[h1->rear] = data;
    int i = h1->rear;
    while(i>0 && h1->data[i]> h1->data[parent(i)]){
        swaph(h1, i,parent(i));
        i = parent(i);
    }

}
void print_heap(heap h1){
    for(int i =0;i<h1.rear;i++){
        printf("%d ",h1.data[i]);
    }
    printf("\n");
}

void heapify(heap * h1){
    if(h1->rear ==0 || h1->rear ==-1) return;
    int i = 0;
    while(i<=(h1->rear)){
        int l = 2*i+1;
        int r = 2*i+2;
        if(l>h1->rear){
            return;
        }
        if(r>=h1->rear){
            if(h1->data[i]<h1->data[l]){
            swaph(h1,i,l);
            }
            return;
        }
        if(h1->data[l]>h1->data[r]){
            swaph(h1,i,l);
            i = l;
        }else{
            swaph(h1,i,r);
            i =r;
        }
    }
}
void delete_max(heap * h1){
    if(h1->rear==-1)return;
    h1->data[0]  = h1->data[h1->rear];
    h1->rear--;
    heapify(h1);
    return;
}
void heap_sort(heap *h1){
    int  k = h1->rear;
```

```
    if(h1->rear == -1) return;
    for(int i = h1->rear;i>0;i++){
        swaph(h1, i,0);
        h1->rear--;
        heapify(h1);
    }
    h1->rear = k;
    print_heap(*h1);
    return;
}
```

}Main.c

```
#include <stdio.h>
#include <stdlib.h>
#include "header.h"


void read_file_to_heap(char *filename, heap *h1) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        perror("Unable to open file");
        exit(1);
    }

    int num;
    while (fscanf(file, "%d", &num) != EOF) {
        insertInheap(h1, num);
    }

    fclose(file);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <filename>\n", argv[0]);
        return 1;
    }

    char *filename = argv[1];
    int maxSize = 100;
    heap h1;
    initHeap(&h1, maxSize);

    read_file_to_heap(filename, &h1);
    printf("Heap before sorting:\n");
    print_heap(h1);

    printf("Heap after sorting:\n");
```

```
    heap_sort(&h1);


    free(h1.data);
    return 0;
}
```

OUTPUT

```
appleApple@Nisargs-MacBook-Air build % cd Heap
appleApple@Nisargs-MacBook-Air Heap % ls
header.h        logic.c         main.c          numbers.txt     out
appleApple@Nisargs-MacBook-Air Heap % gcc main.c logic.c -o out
appleApple@Nisargs-MacBook-Air Heap % ./out numbers.txt
Heap before sorting:
89 67 78 56 23 45 2 12 34
Heap after sorting:
2 12 23 34 45 56 67 78 89
appleApple@Nisargs-MacBook-Air Heap %
```