VISHNU BIBHISHAN GURAV

642403005

ASSIGNMENT 6

header.h

```
typedef struct heap {
   int *h;
   int size;
   int rear;
} heap;


void init_heap(heap *h1, int v);
void insert_heap(heap *h1, int v);
void delete_max(heap *h1);
void print_heap(heap h1);
void heap_sort(heap *h1);
void heapify(heap *h1);
int parent(int index);
void swap(heap *h1, int a, int b);
void readfileandinsert(heap *h1, const char *filename);
```

logic.c

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

void init_heap(heap *h1, int v) {
   h1->h = (int *)malloc(sizeof(int) * v);
   h1->rear = -1;
   h1->size = v;
}
int parent(int index) {
   return (index - 1) / 2;
}

void swap(heap *h1, int a, int b) {
   if (a >= h1->size || b >= h1->size) return;
   int temp = h1->h[a];
   h1->h[a] = h1->h[b];
   h1->h[b] = temp;
}

void insert_heap(heap *h1, int v) {
   if (h1->rear == h1->size - 1) {
      h1->size *= 2;
      h1->h = (int *)realloc(h1->h, sizeof(int) * h1->size);
   }
   h1->rear++;
   h1->h[h1->rear] = v;
   int i = h1->rear;
```

```
    while (i > 0 && h1->h[i] > h1->h[parent(i)]) {
        swap(h1, i, parent(i));
        i = parent(i);
    }
}

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

void heapify(heap *h1) {
    int j = 0;
    while (j <= h1->rear) {
        int lchild = 2 * j + 1;
        int rchild = 2 * j + 2;
        if (lchild > h1->rear) return;
        int max_child = (rchild > h1->rear || h1->h[lchild] > h1->h[rchild]) ? lchild : rchild;
        if (h1->h[j] >= h1->h[max_child]) return;
        swap(h1, j, max_child);
        j = max_child;
    }
}

void delete_max(heap *h1) {
    if (h1->rear == -1) {
        printf("Heap is empty.\n");
```

```
        return;
    }
    h1->h[0] = h1->h[h1->rear--];
    heapify(h1);
}

void heap_sort(heap *h1) {
    int original_size = h1->rear;
    for (int i = h1->rear; i > 0; i--) {
        swap(h1, i, 0);
        h1->rear--;
        heapify(h1);
    }
    h1->rear = original_size;
    print_heap(*h1);
}

void readfileandinsert(heap *h1, const char *filename) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        printf("Failed to open file: %s\n", filename);
        return;
    }
    int num;
    while (fscanf(file, "%d", &num) == 1) {
        insert_heap(h1, num);
    }
    fclose(file);
}
```

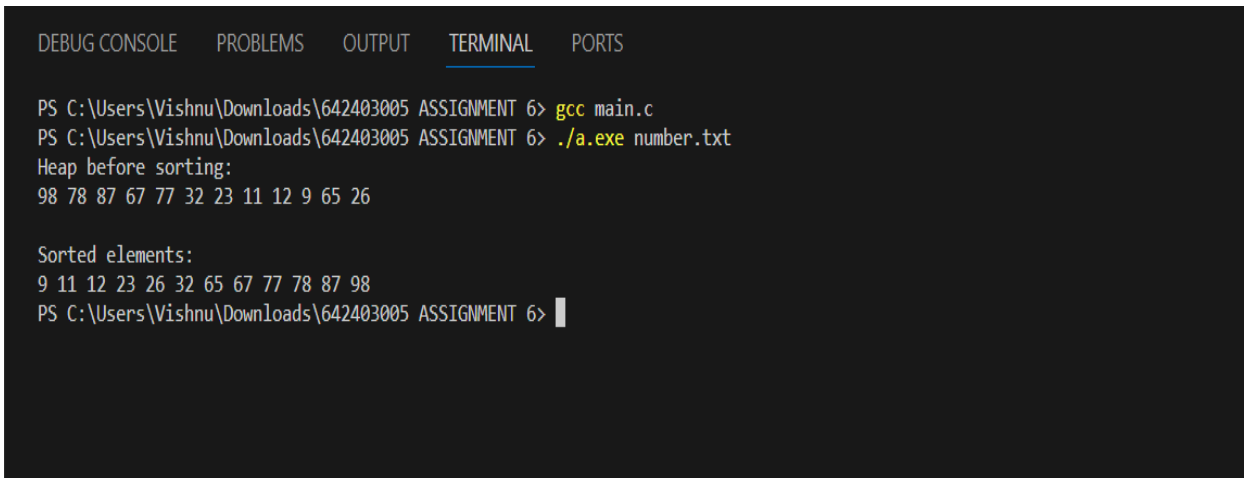
main.c

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

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

   heap h1;
   init_heap(&h1, 10);

   readfileandinsert(&h1, argv[1]);

   printf("Heap before sorting:\n");
   print_heap(h1);

   printf("\nSorted elements:\n");
   heap_sort(&h1);

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

num.txt

87 65 32 11 77 98 23 12 67 9 78 26

Output


DEBUG CONSOLE
PROBLEMS
OUTPUT
TERMINAL
PORTS
PS C:\Users\Vishnu\Downloads\642403005 ASSIGNMENT 6> gcc main.c
PS C:\Users\Vishnu\Downloads\642403005 ASSIGNMENT 6> ./a.exe number.txt
Heap before sorting:
98 78 87 67 77 32 23 11 12 9 65 26
Sorted elements:
9 11 12 23 26 32 65 67 77 78 87 98
PS C:\Users\Vishnu\Downloads\642403005 ASSIGNMENT 6>