Heap sort(Max size)

#include <stdio.h>

#define MAX\_SIZE 100

// Function prototype for adjust

void adjust(int arr[], int i, int n);

void insert(int arr[], int \*n, int item) {

int i = \*n;

(\*n)++;

while (i > 1 && arr[i / 2] < item) {

arr[i] = arr[i / 2];

i /= 2;

}

arr[i] = item;

}

int deleteMax(int arr[], int \*n) {

if (\*n == 0) {

printf("Heap is empty\n");

return -1; // return an error value

}

int max = arr[1];

arr[1] = arr[\*n];

(\*n)--;

adjust(arr, 1, \*n);

return max;

}

void adjust(int arr[], int i, int n) {

int item = arr[i];

int j = 2 \* i;

while (j <= n) {

if (j < n && arr[j] < arr[j + 1]) {

j = j + 1; // Select the larger child

}

if (item >= arr[j]) {

break; // A position for item is found

}

arr[i] = arr[j];

i = j;

j = 2 \* i;

}

arr[i] = item;

}

void displayHeap(int arr[], int n) {

for (int i = 1; i <n; i++) {

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

int arr[MAX\_SIZE], n = 1, choice, element;

while (1) {

printf("1. Insert\n2. Delete Max\n3. Display Heap\n4. Exit\nEnter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter element to insert: ");

scanf("%d", &element);

insert(arr, &n, element);

break;

case 2:

element = deleteMax(arr, &n);

if (element != -1) {

printf("Deleted max element: %d\n", element);

}

break;

case 3:

printf("Heap: ");

displayHeap(arr, n);

break;

case 4:

return 0;

default:

printf("Invalid choice\n");

}

}

return 0;

}

**Sample Output:**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 1**

**Enter element to insert: 12**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 1**

**Enter element to insert: 23**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 1**

**Enter element to insert: 45**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 3**

**Heap: 45 12 23**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 1**

**Enter element to insert: 56**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 3**

**Heap: 56 45 23 12**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 2**

**Deleted max element: 56**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 3**

**Heap: 45 12 23**

**1. Insert**

**2. Delete Max**

**3. Display Heap**

**4. Exit**

**Enter your choice: 4**

**=== Code Execution Successful ===**