**Assignment – 5**

**Q1. Implementation of Heap Sort and display the working of it.**

#include <stdio.h>  
#include <stdlib.h>  
  
void heap\_sort(int\*, int);  
void build\_max\_heap(int\*);  
void max\_heapify(int\*, int);  
int\* create\_array(int);  
void display\_array(int\*, int);  
  
int heap\_size, max\_heapify\_call = 0;  
  
int main() {  
 int size;  
 printf("Enter the size of the array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 heap\_sort(array, size);  
 printf("\nThe array elements after sorted are: ");  
 display\_array(array, size);  
 free(array);  
 return 0;  
}  
  
void heap\_sort(int\* array, int size) {  
 int i;  
 heap\_size = size;  
 build\_max\_heap(array);  
  
 for (i = size - 1; i > 0; i--) {  
 int temp = array[0];  
 array[0] = array[i];  
 array[i] = temp;  
 printf("The partial sorted array at i=%d: ", i);  
 display\_array(array, size);  
 heap\_size--;  
 printf("\nMax Heapify call from heap sort for i=%d\n", i);  
 max\_heapify(array, 0);  
 }  
}  
  
void build\_max\_heap(int\* array) {  
 int i;  
 printf("\nBuild Max Heap call on array: ");  
 display\_array(array, heap\_size);  
  
 for (i = heap\_size / 2 - 1; i >= 0; i--) {  
 max\_heapify(array, i);  
 }  
}  
  
void max\_heapify(int\* array, int i) {  
 printf("Max Heapify call %d on array for i=%d i.e. A[i]=%d\n", ++max\_heapify\_call, i, array[i]);  
 printf("Array before: ");  
 display\_array(array, heap\_size);  
 int left = (i + 1) \* 2 - 1;  
 int right = (i + 1) \* 2;  
 int max = i;  
  
 if (left < heap\_size && array[left] > array[max]) {  
 max = left;  
 }  
 if (right < heap\_size && array[right] > array[max]) {  
 max = right;  
 }  
 if (max != i) {  
 int temp = array[i];  
 array[i] = array[max];  
 array[max] = temp;  
 printf("Array after: ");  
 display\_array(array, heap\_size);  
  
 if (max < heap\_size / 2) {  
 max\_heapify(array, max);  
 }  
 } else {  
 printf("Array after: ");  
 display\_array(array, heap\_size);  
 }  
}  
  
int\* create\_array(int size) {  
 int \*array = (int\*)malloc(sizeof(int) \* size), i;  
  
 if (!array) {  
 printf("Memory was not allocated");  
 exit(0);  
 }  
 printf("Enter the array elements: ");  
  
 for (i = 0; i < size; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
void display\_array(int\* array, int size) {  
 int i;  
  
 for (i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n");  
}

**Output**:

Enter the size of the array: 10  
Enter the array elements: 4 1 3 2 16 9 10 14 8 7  
  
Build Max Heap call on array: 4 1 3 2 16 9 10 14 8 7   
Max Heapify call 1 on array for i=4 i.e. A[i]=16  
Array before: 4 1 3 2 16 9 10 14 8 7   
Array after: 4 1 3 2 16 9 10 14 8 7   
Max Heapify call 2 on array for i=3 i.e. A[i]=2  
Array before: 4 1 3 2 16 9 10 14 8 7   
Array after: 4 1 3 14 16 9 10 2 8 7   
Max Heapify call 3 on array for i=2 i.e. A[i]=3  
Array before: 4 1 3 14 16 9 10 2 8 7   
Array after: 4 1 10 14 16 9 3 2 8 7   
Max Heapify call 4 on array for i=1 i.e. A[i]=1  
Array before: 4 1 10 14 16 9 3 2 8 7   
Array after: 4 16 10 14 1 9 3 2 8 7   
Max Heapify call 5 on array for i=4 i.e. A[i]=1  
Array before: 4 16 10 14 1 9 3 2 8 7   
Array after: 4 16 10 14 7 9 3 2 8 1   
Max Heapify call 6 on array for i=0 i.e. A[i]=4  
Array before: 4 16 10 14 7 9 3 2 8 1   
Array after: 16 4 10 14 7 9 3 2 8 1   
Max Heapify call 7 on array for i=1 i.e. A[i]=4  
Array before: 16 4 10 14 7 9 3 2 8 1   
Array after: 16 14 10 4 7 9 3 2 8 1   
Max Heapify call 8 on array for i=3 i.e. A[i]=4  
Array before: 16 14 10 4 7 9 3 2 8 1   
Array after: 16 14 10 8 7 9 3 2 4 1   
The partial sorted array at i=9: 1 14 10 8 7 9 3 2 4 16   
  
Max Heapify call from heap sort for i=9  
Max Heapify call 9 on array for i=0 i.e. A[i]=1  
Array before: 1 14 10 8 7 9 3 2 4   
Array after: 14 1 10 8 7 9 3 2 4   
Max Heapify call 10 on array for i=1 i.e. A[i]=1  
Array before: 14 1 10 8 7 9 3 2 4   
Array after: 14 8 10 1 7 9 3 2 4   
Max Heapify call 11 on array for i=3 i.e. A[i]=1  
Array before: 14 8 10 1 7 9 3 2 4   
Array after: 14 8 10 4 7 9 3 2 1   
The partial sorted array at i=8: 1 8 10 4 7 9 3 2 14 16   
  
Max Heapify call from heap sort for i=8  
Max Heapify call 12 on array for i=0 i.e. A[i]=1  
Array before: 1 8 10 4 7 9 3 2   
Array after: 10 8 1 4 7 9 3 2   
Max Heapify call 13 on array for i=2 i.e. A[i]=1  
Array before: 10 8 1 4 7 9 3 2   
Array after: 10 8 9 4 7 1 3 2   
The partial sorted array at i=7: 2 8 9 4 7 1 3 10 14 16   
  
Max Heapify call from heap sort for i=7  
Max Heapify call 14 on array for i=0 i.e. A[i]=2  
Array before: 2 8 9 4 7 1 3   
Array after: 9 8 2 4 7 1 3   
Max Heapify call 15 on array for i=2 i.e. A[i]=2  
Array before: 9 8 2 4 7 1 3   
Array after: 9 8 3 4 7 1 2   
The partial sorted array at i=6: 2 8 3 4 7 1 9 10 14 16   
  
Max Heapify call from heap sort for i=6  
Max Heapify call 16 on array for i=0 i.e. A[i]=2  
Array before: 2 8 3 4 7 1   
Array after: 8 2 3 4 7 1   
Max Heapify call 17 on array for i=1 i.e. A[i]=2  
Array before: 8 2 3 4 7 1   
Array after: 8 7 3 4 2 1   
The partial sorted array at i=5: 1 7 3 4 2 8 9 10 14 16

Max Heapify call from heap sort for i=5  
Max Heapify call 18 on array for i=0 i.e. A[i]=1  
Array before: 1 7 3 4 2   
Array after: 7 1 3 4 2   
Max Heapify call 19 on array for i=1 i.e. A[i]=1  
Array before: 7 1 3 4 2   
Array after: 7 4 3 1 2   
The partial sorted array at i=4: 2 4 3 1 7 8 9 10 14 16   
  
Max Heapify call from heap sort for i=4  
Max Heapify call 20 on array for i=0 i.e. A[i]=2  
Array before: 2 4 3 1   
Array after: 4 2 3 1   
Max Heapify call 21 on array for i=1 i.e. A[i]=2  
Array before: 4 2 3 1   
Array after: 4 2 3 1   
The partial sorted array at i=3: 1 2 3 4 7 8 9 10 14 16   
  
Max Heapify call from heap sort for i=3  
Max Heapify call 22 on array for i=0 i.e. A[i]=1  
Array before: 1 2 3   
Array after: 3 2 1   
The partial sorted array at i=2: 1 2 3 4 7 8 9 10 14 16   
  
Max Heapify call from heap sort for i=2  
Max Heapify call 23 on array for i=0 i.e. A[i]=1  
Array before: 1 2   
Array after: 2 1   
The partial sorted array at i=1: 1 2 3 4 7 8 9 10 14 16   
  
Max Heapify call from heap sort for i=1  
Max Heapify call 24 on array for i=0 i.e. A[i]=1  
Array before: 1   
Array after: 1   
  
The array elements after sorted are: 1 2 3 4 7 8 9 10 14 16