**10665404**

#include <stdio.h>

#include <stdlib.h>

int number = 0;

int flag = 0;

/\*---------------------------------------Function for Printing Stars----------------------------------------------\*/

void print\_star(int array[])

{

int first\_count1,second\_count2;

printf("\n");

first\_count1 = 0;

while(first\_count1 < number)

{

printf("Array Value - %d : ",array[first\_count1]);

second\_count2 = 0;

while (second\_count2<array[first\_count1])

{

printf("\*"); //Printing stars

second\_count2++;

}

first\_count1++;

printf("\n");

}

}

/\*----------------------------------------------------------------------------------------------------------------\*/

/\*---------------------------------------------Swapping Function--------------------------------------------------\*/

void swapping(int \*left, int \*right)

{

int temp = \*left;

\*left = \*right;

\*right = temp;

}

/\*----------------------------------------------------------------------------------------------------------------\*/

/\*---------------------------------------------Merge Sort--------------------------------------------------------\*/

void msort(int array[],int lower,int middle,int higher) //function for main sorting elements

{

int j,n,p,x,other[1000];

x=lower;

j=lower;

n=middle+1;

while((n<=higher) && (x<=middle))

{

if(array[n] >= array[x]){

other[j] = array[x];

x++;

}

else{

other[j] = array[n];

n++;

}

j++;

}

if(x > middle){

p=n;

while(p<=higher){

other[j] = array[p];

j++;

p++;

}

}

else{

p=x;

while(p<=middle){

other[j] = array[p];

j++;

p++;

}

}

p = lower;

while(p <= higher){

array[p] = other[p];

p++;

}

if ( number <= 20){

print\_star(array);

}

}

void merge\_sort(int array[],int lower,int higher)

{

int middle;

if(higher > lower){

middle = (lower+higher)/2;

merge\_sort(array,lower,middle);

merge\_sort(array,middle+1,higher);

msort(array,lower,middle,higher);

}

}

/\*--------------------------------------------------------------------------------------------------------------\*/

/\*------------------------------------------Counting Sort -----------------------------------------------------\*/

void counting\_sort(int array[], int number)

{

int i;

int maximum = -1;

i = 0;

while(i < number){

if(array[i] > maximum){

maximum = array[i];

}

i++;

}

int second\_array[maximum+1];

i = 0;

while(i <= maximum){

second\_array[i] = 0;

i++;

}

i = 0;

while(i < number){

int second = array[i];

second\_array[second] = second\_array[second] + 1;

i++;

}

printf("\nMiddle Array : ");

i = 0;

while(i <= maximum){

printf("%d ", second\_array[i]); //printing middle array

i++;

}

printf("\n");

i = 1;

while(i <= maximum){

second\_array[i] = second\_array[i] + second\_array[i-1];

i++;

}

int result[number];

i = 0;

while(i < number){

int second = array[i];

int location = second\_array[second] - 1;

result[location] = second;

second\_array[second] = second\_array[second] - 1;

i++;

}

i = 0;

while(i < number){

array[i] = result[i];

i++;

}

}

/\*-----------------------------------------------------------------------------------------------------------\*/

int main()

{

int selection,count,i;

int array[1000],sorted\_array[1000];

int max\_element;

printf("Select the Sorting Algorithm:: \n");

printf("1. Welcome\n");

printf("2. Thank you for using my code \n");

printf("3. Merge Sort (Enter element greater than 0) \n");

printf("\nEnter your choice::");

scanf("%d",&selection);

if(selection==2){

flag=1;

}

while(number < 1 || number > 1001){

printf("\nEnter the value of Numbers between 1 and 1000::");

scanf ("%d",&number);

}

if (number <= 20){

srand(time(NULL));

count = 0;

while(count < number){

array[count]= rand() % 15 + 1; //if value is under 20 then randomized numbers within 15

count++;

}

}

else{

srand(time(NULL));

count = 0;

while(count < number){

if (flag == 1){

array[count]= rand() % 98 + 1; //for counting sort, randomized number between 99

}

else{

array[count]= rand() % 1000 + 1; //for value exceeding 20, randomized numbers within 1000

}

count++;

}

}

printf("\nThe Random Numbers are:- ");

count = 0;

while(count < number){

printf("%d ",array[count]);

count++;

}

printf("\n\n");

switch(selection)

{

case 2: //Case 2 for Counting Sort

if ( number <= 20){

print\_star(array);

}

counting\_sort(array, number);

if ( number <= 20){

print\_star(array);

}

printf("\nThe sorted array is:: ");

i = 0;

while(i < number)

{

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

i++;

}

printf("\n");

break;

case 3: //Case 3 for Merge Sort

if (number <=20){

print\_star(array);

}

merge\_sort(array,0,number);

printf("\nThe sorted array is:: ");

i = 0;

while(i < number)

{

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

i++;

}

printf("\n");

break;

default:

printf("You Entered a Wrong Choice\n");

break;

}

return 0;

}

