**Data Structures in C LAB – {4}**

Akshaya agarwal

2247207

3-MCA B

**CODE :**

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

int a[20];

int n;

int random() {

int i=0;

printf("\nEnter the range of the array\n");

scanf("%d", &n);

for(i=0; i<n; i++) {

a[i]=rand()%100;

}

printf("\nThe elements of array are\n");

for(i=0; i<n; i++) {

printf("\nThe elements at a[%d] = %d", i,a[i]);

}

return n;

}

int manual() {

int i=0;

printf("\nEnter the length of array\n");

scanf("%d", &n);

for(i=0; i<n; i++) {

printf("\nEnter the element for a[%d] : ", i);

scanf("%d", &a[i]);

}

printf("\nThe elements your entered are\n");

for(i=0; i<n; i++) {

printf("\nThe elements at a[%d] = %d", i,a[i]);

}

return n;

}

void merge(int arr[], int left, int mid, int right) {

int n1 = mid - left + 1, i, j;

int n2 = right - mid;

int L[n1], R[n2];

for (i = 0; i < n1; i++)

{

L[i] = arr[left + i];

}

for (j = 0; j < n2; j++)

{

R[j] = arr[mid + 1 + j];

}

i = 0;

j = 0;

int k = left;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergesort(int arr[], int left, int right)

{

if (left < right)

{

int mid = left + (right - left) / 2, i;

mergesort(arr, left, mid);

mergesort(arr, mid + 1, right);

printf("Iteration result: ");

for (i = left; i <= right; i++)

{

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

}

printf("\n");

merge(arr, left, mid, right);

}

}

void printarray(int a[], int n)

{

int i;

for (i = 0; i < n; i++)

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

printf("\n");

}

int main() {

int n,i,j,k,ch;

while(1) {

printf("\nPlease select your choice from the options below\n");

printf("\n1.Enter the input randomly.\n2.Enter the input manually\n3.Print the sorted array.\n4.Print the array\n5.Exit\n");

while (scanf("%d", &ch) != 1) {

printf("Incorrect choice. Enter an integer number.\n");

while (getchar() != '\n');

}

printf("--------------------------------------\n");

switch(ch) {

case 1:

n=random();

break;

case 2:

n=manual();

break;

case 3: mergesort(a,0,n-1);

//for(i=0;i<n;i++)

//{

// printf("\n%d\n", a[i]);

//}

break;

case 4: printarray(a,n);

break;

case 5:exit(0);

default:

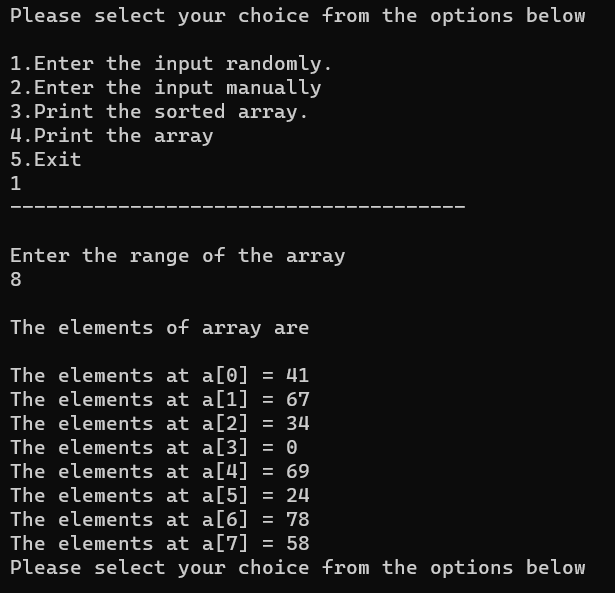
printf("Wrong choice");

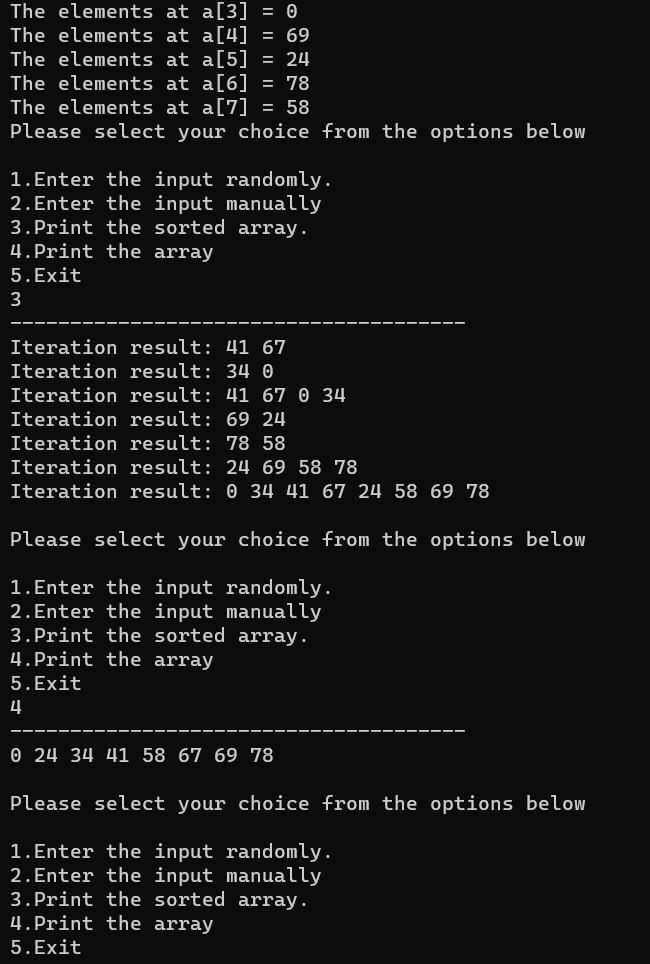
}

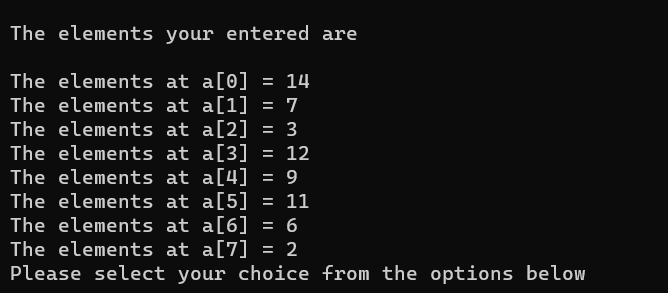
}

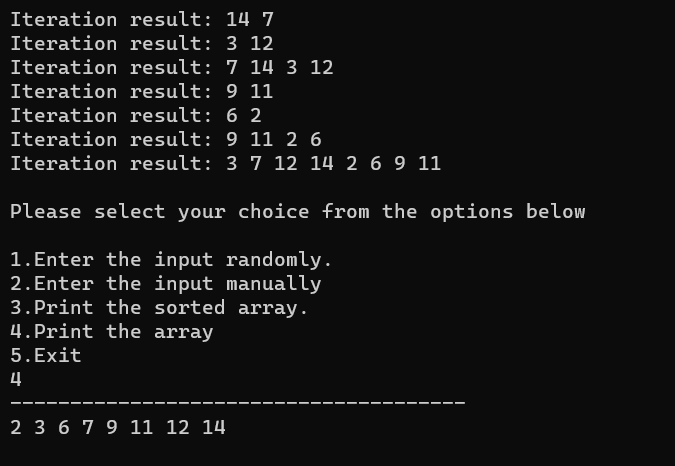
}

**OUTPUT :**

****

****

****

****