* **BUBBLE SORT**

**SOURC CODE:**

#include<stdio.h>

void printarray(int \*a, int n) {

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

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

}

printf("\n");

}

void bubblesort(int \*a, int n) {

int temp;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - 1 - i; j++) {

if (a[j] > a[j + 1]) {

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

}

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int a[n];

printf("Enter %d elements for the array:\n", n);

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

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

}

printf("Original array: ");

printarray(a, n);

bubblesort(a, n);

printf("Sorted array: ");

printarray(a, n);

return 0;

}

* **INSERTION SORT**

**SOURCE CODE:**

#include<stdio.h>

void printarray(int \*a ,int n){

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

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

}

printf("\n");

}

void insertionsort(int \*a,int n){

int key , j ;

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

key = a[i];

j = i-1;

while(j>=0 && a[j]>key){

a[j+1]=a[j];

j--;

}

a[j+1]=key;

}

}

int main()

{

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int a[n];

printf("Enter %d elements for the array:\n", n);

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

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

}

printarray(a,n);

insertionsort(a,n);

printarray(a,n);

return 0;

}

* **SELECTION SORT**

**SOURCE CODE:**

#include<stdio.h>

void printArray(int \*a, int n){

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

{

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

}

printf("\n");

}

void selectionSort(int \*a, int n){

int indexOfMin , temp;

printf("Running Selection sort...\n");

for (int i = 0; i < n-1; i++)

{

indexOfMin = i;

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

{

if(a[j] < a[indexOfMin]){

indexOfMin = j;

}

}

// Swap A[i] and A[indexOfMin]

temp = a[i];

a[i] = a[indexOfMin];

a[indexOfMin] = temp;

}

}

int main(){

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int a[n];

printf("Enter %d elements for the array:\n", n);

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

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

}

printArray(a, n);

selectionSort(a, n);

printArray(a, n);

return 0;

}

* **MERGE SORT**

**SOURCE CODE:**

#include <stdio.h>

void printArray(int \*a, int n)

{

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

{

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

}

printf("\n");

}

void merge(int a[], int mid, int low, int high)

{

int i, j, k, B[100];

i = low;

j = mid + 1;

k = low;

while (i <= mid && j <= high)

{

if (a[i] < a[j])

{

B[k] = a[i];

i++;

k++;

}

else

{

B[k] = a[j];

j++;

k++;

}

}

while (i <= mid)

{

B[k] = a[i];

k++;

i++;

}

while (j <= high)

{

B[k] = a[j];

k++;

j++;

}

// for printing the b[] into a[]

for (int i = low; i <= high; i++)

{

a[i] = B[i];

}

}

void mergeSort(int a[], int low, int high){

int mid;

if(low<high){

mid = (low + high) /2;

mergeSort(a, low, mid);

mergeSort(a, mid+1, high);

merge(a, mid, low, high);

}

}

int main()

{

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int a[n];

printf("Enter %d elements for the array:\n", n);

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

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

}

printArray(a, n);

mergeSort(a, 0, 6);

printArray(a, n);

return 0;

}