Merge sort:

#include<iostream>

using namespace std;

void swapping(int &a, int &b) {

int temp = a;

a = b;

b = temp;

}

void display(int \*array, int size) {

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

cout << array[i] << " ";

cout << endl;

}

void merge(int \*array, int l, int m, int r) {

int i, j, k, nl, nr;

nl = m-l+1; nr = r-m;

int larr[nl], rarr[nr];

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

larr[i] = array[l+i];

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

rarr[j] = array[m+1+j];

i = 0; j = 0; k = l;

while(i < nl && j<nr) {

if(larr[i] <= rarr[j]) {

array[k] = larr[i];

i++;

}else{

array[k] = rarr[j];

j++;

}

k++;

}

while(i<nl) {

array[k] = larr[i];

i++; k++;

}

while(j<nr) {

array[k] = rarr[j];

j++; k++;

}

}

void mergeSort(int \*array, int l, int r) {

int m;

if(l < r) {

int m = l+(r-l)/2;

mergeSort(array, m+1, r);

merge(array, l, m, r);

}

}

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

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

cin >> arr[i];

}

cout << "Array before Sorting: ";

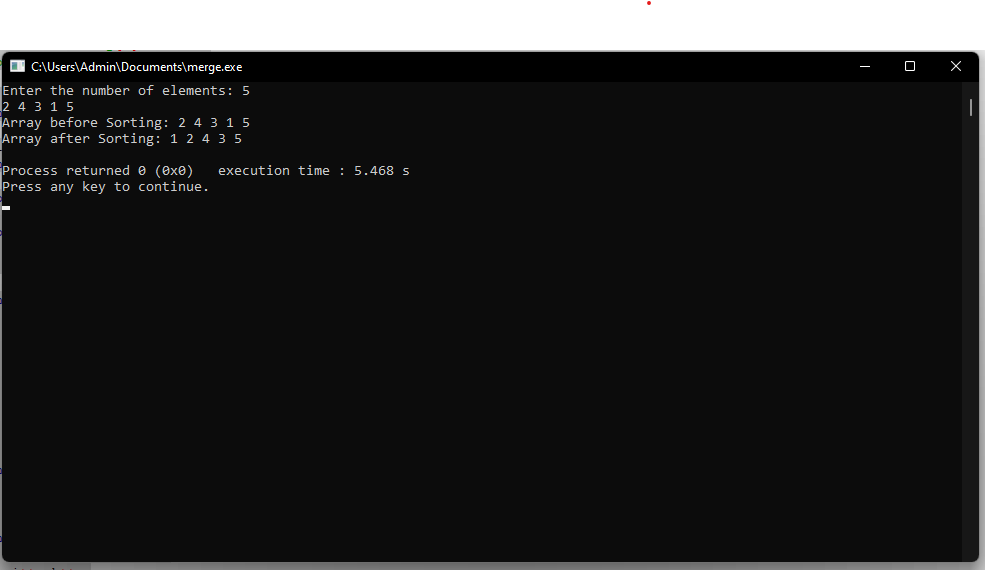
display(arr, n);

mergeSort(arr, 0, n-1);

cout << "Array after Sorting: ";

display(arr, n);

}



Quick sort:

#include<iostream>

using namespace std;

void swap(int arr[] , int pos1, int pos2){

int temp;

temp = arr[pos1];

arr[pos1] = arr[pos2];

arr[pos2] = temp;

}

int partition(int arr[], int low, int high, int pivot){

int i = low;

int j = low;

while( i <= high){

if(arr[i] > pivot){

i++;

}

else{

swap(arr,i,j);

i++;

j++;

}

}

return j-1;

}

void quickSort(int arr[], int low, int high){

if(low < high){

int pivot = arr[high];

int pos = partition(arr, low, high, pivot);

quickSort(arr, low, pos-1);

quickSort(arr, pos+1, high);

}

}

int main()

{

int n ;

cout << "enter the size of array";

cin>>n;

int arr[n];

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

cin>>arr[i];

}

quickSort(arr, 0 , n-1);

cout<<"The sorted array is: ";

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

cout<< arr[i]<<" ";

}

}

