**TUGAS PRAKTIKUM STRUKTUR DATA**

**JOBSHEET 8**

****

**Dosen Pengampu:**

**Randi Proska Sandra, S.Pd, M.Sc**

**Kode Kelas:**

**202323430158**

**Disusun Oleh:**

**Wahyu Abdil Afif**

**23343085**

**PROGRAM STUDI INFORMATIKA (NK)**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2024**

1. Bubble Sort:

Source code:

// created by wahyu abdil afif\_23343085

#include <stdio.h>

void bubbleSort(int arr[], int n)

{

int i, j, temp;

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

{

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

{

if (arr[j] > arr[j+1])

{

temp = arr[j];

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

arr[j+1] = temp;

}

}

}

}

int main()

{

int arr[] = {17, 31, 25, 12, 22, 11, 3};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

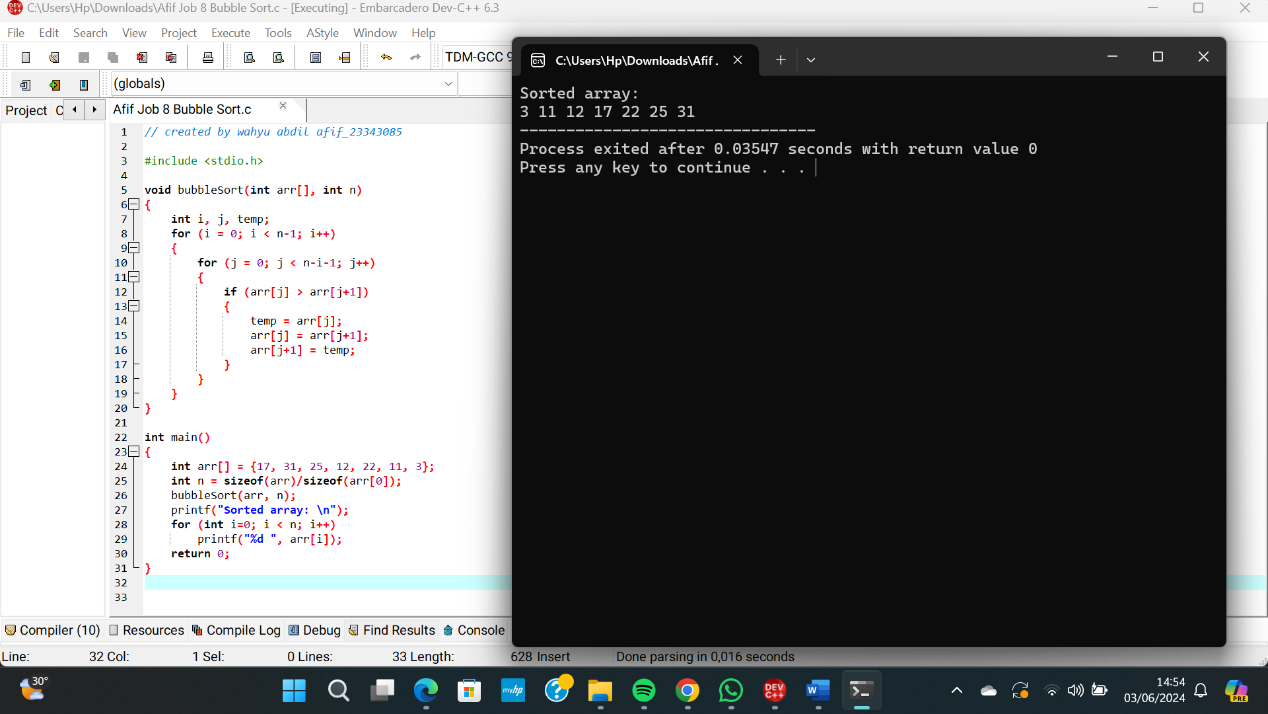
printf("Sorted array: \n");

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

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

return 0;

}



Penjelasan:

* Algoritma bubble sort bekerja dengan membandingkan elemen-elemen yang berdekatan dalam array.
* Jika elemen pertama lebih besar daripada elemen kedua, maka elemen-elemen tersebut ditukar.
* Proses ini diulang sampai semua elemen diurutkan.
* Prinsip utama bubble sort adalah menggelembungkan elemen terbesar ke atas array seperti gelembung dalam air.
* Bubble sort adalah pilihan yang baik untuk mengurutkan array kecil atau untuk tujuan pembelajaran karena kesederhanaannya. Namun, untuk array besar, disarankan menggunakan algoritma pengurutan yang lebih efisien seperti merge sort atau quicksort.

1. Insertion sort:

Source code:

// created by wahyu abdil afif\_23343085

#include <stdio.h>

void insertionSort(int arr[], int n)

{

int i, key, j;

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

{

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key)

{

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

j = j - 1;

}

arr[j + 1] = key;

}

}

int main()

{

int arr[] = {17, 31, 25, 12, 22, 11, 3};

int n = sizeof(arr)/sizeof(arr[0]);

insertionSort(arr, n);

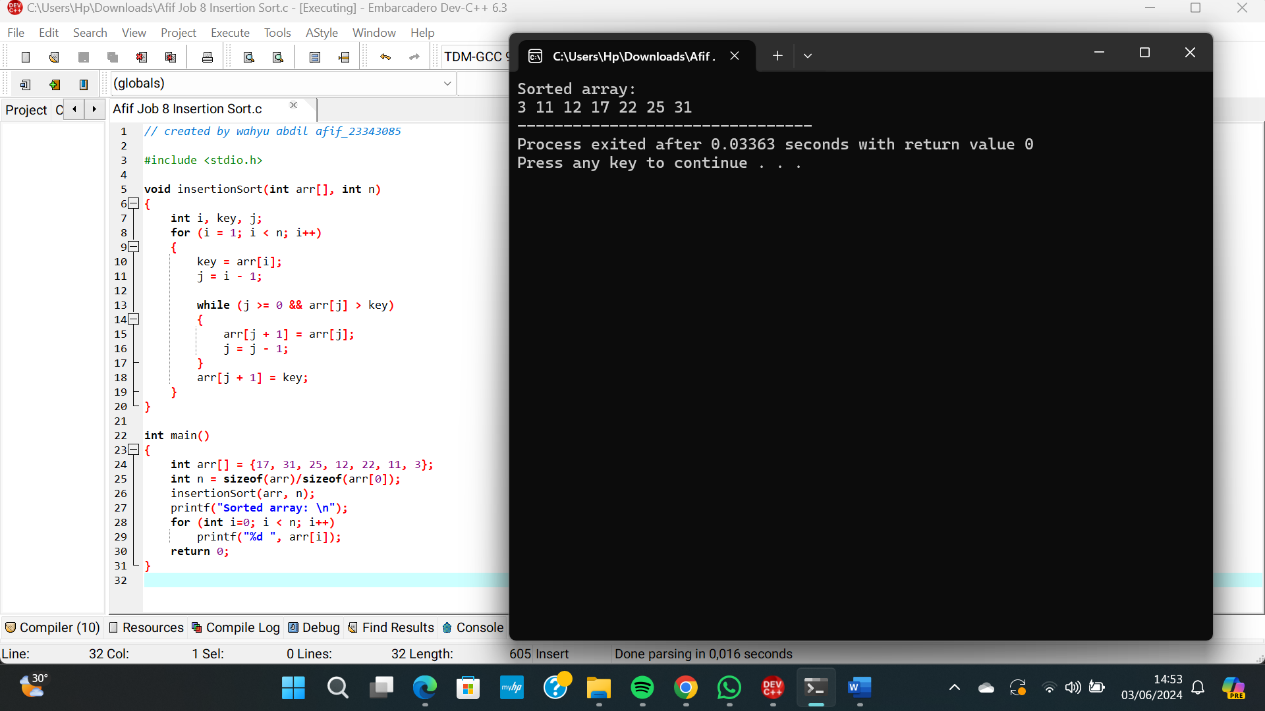
printf("Sorted array: \n");

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

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

return 0;

}



Penjelasan:

* Algoritma insertion sort bekerja dengan memasukkan elemen satu per satu ke dalam array yang telah diurutkan.
* Elemen baru dibandingkan dengan elemen-elemen yang telah diurutkan sebelumnya.
* Jika elemen baru lebih kecil daripada elemen yang telah diurutkan, maka elemen-elemen yang telah diurutkan digeser ke kanan untuk memberikan ruang bagi elemen baru.
* Proses ini diulang sampai semua elemen dimasukkan ke dalam array.
* Prinsip utama insertion sort adalah menyisipkan elemen-elemen ke dalam array yang telah diurutkan seperti memasukkan kartu ke dalam tumpukan kartu yang sudah diurutkan.
* Insertion sort adalah pilihan yang baik untuk data berukuran kecil atau untuk skenario di mana kesederhanaan dan kemudahan implementasi menjadi prioritas. Namun, untuk data berukuran besar, disarankan menggunakan algoritma sorting yang lebih efisien seperti merge sort atau quicksort.