



Facultad de Ingeniería UNAM



Maestro: Tista Garcia Edgar MI.

Asignatura: Estructura de Datos y Algoritmos II

Práctica 1: Bubble Sort & Quick Sort

Alumno: Mejía Alba Israel Hipólito

Grupo:8

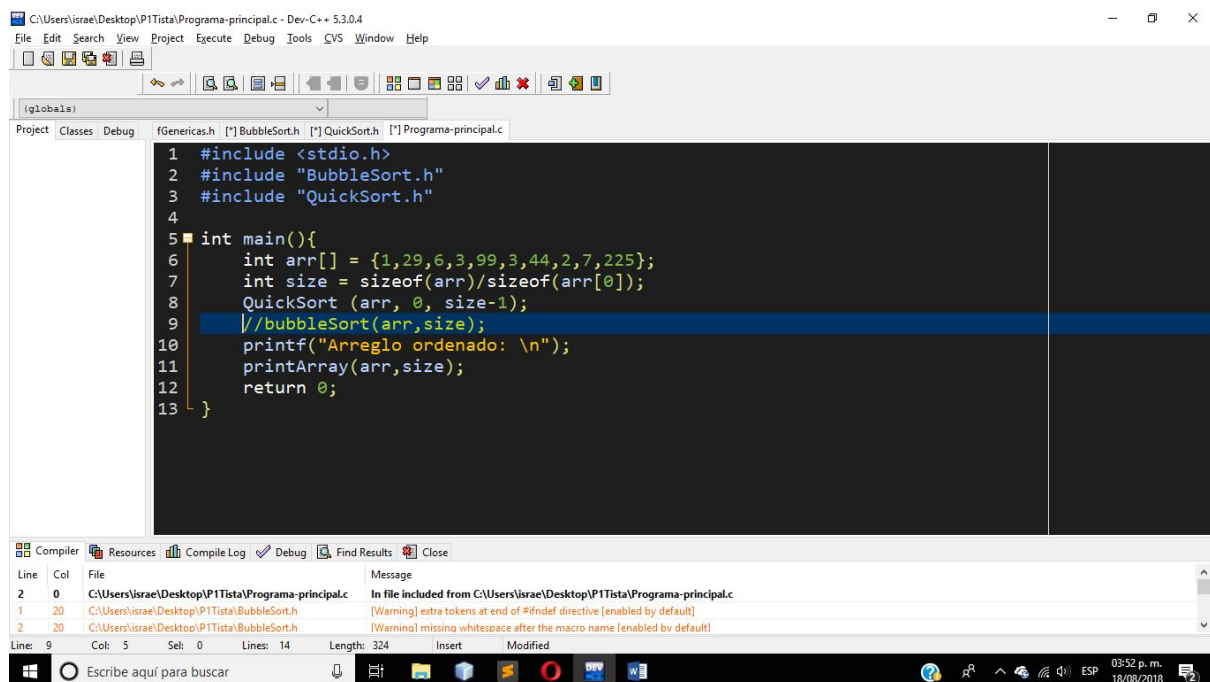
Objetivo: El estudiante identificará la estructura de los algoritmos de ordenamiento Bubble Sort y Quick Sort.

Desarrollo:

Actividades:

1.1.-Verificar el Funcionamiento de Quicksort y Bubble Sort e incluye captura de pantalla de cada uno, comenta lo que te parezca relevante:

Función principal: Aquí se declara el arreglo a ordenar, se verifica su tamaño y se mandan a llamar las funciones necesarias para su ordenamiento.



```
1 #include <stdio.h>
2 #include "BubbleSort.h"
3 #include "QuickSort.h"
4
5 int main(){
6     int arr[] = {1,29,6,3,99,3,44,2,7,225};
7     int size = sizeof(arr)/sizeof(arr[0]);
8     QuickSort (arr, 0, size-1);
9     //bubbleSort(arr,size);
10    printf("Arreglo ordenado: \n");
11    printArray(arr,size);
12    return 0;
13 }
```

Archivo f.Genéricas:

Contiene el código de intercambios de las variables de nuestro arreglo y de igual manera esta la funcion para imprimir cada uno de los elementos del arreglo.

```
1 #ifndef _fGenericas.h
2 #define _fGenericas.h
3
4 void swap(int *a, int *b){
5     int t = *a;
6     *a = *b;
7     *b = t;
8 }
9
10 void printArray(int arr[],int size){
11     int i;
12     for (i=0; i < size; i++)
13         printf("%d ", arr[i]);
14     printf("\n");
15 }
16 #endif
```

Archivo QuickSort.h :

En este archivo tenemos tres funciones necesarias para realizar el algoritmo Quicksort, está la que imprime a los subarreglos, la que realiza las particiones necesarias y la función QuickSort que manda a llamar las funciones anteriormente solicitadas para poder ordenar a nuestro arreglo

```
1 #ifndef _QuickSort.h
2 #define _QuickSort.h
3 #include "fGenericas.h"
4
5 void printSubArray (int arr[], int low, int high){
6     int i;
7     for (i=low;i<=high;i++)
8         printf("%d ", arr[i]);
9     printf("\n");
10 }
11
12 int partition (int arr[],int low, int high){
13     int pivot = arr[high];
14     printf("Pivote: %d \n ", pivot);
15     int j,i = (low - 1);
16     for (j = low; j <= high - 1; j++){
17         if (arr[j] <= pivot){
18             i++;
19             swap(&arr[i],&arr[j]);
20         }
21     }
22 }
```

```
21 }
22 swap(&arr[i+1], &arr[high]);
23 return (i+1);
24 }
25
26 void QuickSort(int arr[],int low, int high){
27     if(low < high){
28         int pi = partition(arr, low, high);
29         printf("Sub array 1 (de %d): \t ",arr[pi]);
30         printSubArray(arr,low,pi-1);
31         QuickSort(arr,low,pi-1);
32         printf("Sub array 2: (de %d): \t ",arr[pi]);
33         printSubArray(arr,pi+1,high);
34         QuickSort(arr,pi + 1, high);
35     }
36 }
37
38 #endif
39
40
```

Compiler Resources Compile Log Debug Find Results Close

Line	Col	File	Message
2	0	C:\Users\israel\Desktop\P1Tista\Programa-principal.c	In file included from C:\Users\israel\Desktop\P1Tista\Programa-principal.c
1	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] extra tokens at end of #ifndef directive [enabled by default]
2	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] missing whitespace after the macro name [enabled by default]

Line: 40 Col: 1 Sel: 0 Lines: 40 Length: 961 Insert Modified

Escribe aquí para buscar

04:03 p. m. 18/08/2018

Archivo BubbleSort.h: En este archivo se realiza la asignación de la variable índice, así como el recorrido del mismo en el arreglo y los intercambios necesarios para acomodar el mismo.

```
1 #ifndef _BubbleSort.h
2 #define _BubbleSort.h
3 #include "fGenericas.h"
4 void bubbleSort(int a[], int size){
5     int i,j;
6     int index;
7     i= size-1;
8     while (i>0){
9         index = 0;
10        for(j=0;j<i;j++){
11            if(a[j+1]<a[j]){
12                swap(&a[j+1],&a[j]);
13                index=j;
14            }
15            printArray(a,size);
16        }
17        printf("Fin Iteracion \n");
18        i=index;
19    }
20 }
21 #endif
```

Compiler Resources Compile Log Debug Find Results Close

Line	Col	File	Message
2	0	C:\Users\israel\Desktop\P1Tista\Programa-principal.c	In file included from C:\Users\israel\Desktop\P1Tista\Programa-principal.c
1	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] extra tokens at end of #ifndef directive [enabled by default]
2	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] missing whitespace after the macro name [enabled by default]

Line: 4 Col: 1 Sel: 0 Lines: 21 Length: 447 Insert Done parsing

Escribe aquí para buscar

04:08 p. m. 18/08/2018

Ejecución de QuickSort:

Se puede ver como en se imprime el valor del pivote con el paso en el que se va ordenando el arreglo, así como ambos subarreglos y finalmente el arreglo ordenado.

```
#ifndef QuickSort_h
#define QuickSort_h
#include "fGenerics.h"

void printSubarray (int arr[], int low, int high){
    int i;
    for (i=low; i<=high; i++){
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int partition (int arr[], int low, int high){
    int pivot = arr[high];
    printf("Pivote: %d\n", pivot);
    int j = (low - 1);
    for (i = low; i <= high - 1; i++){
        if (arr[i] <= pivot){
            j++;
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[j+1], arr[high]);
    return (j+1);
}

void QuickSort (int arr[], int low, int high){
    if (low < high){
        int pi = partition(arr, low, high);
        printf("Sub array 1 (de %d): ", arr[pi]);
        printSubarray(arr, low, pi-1);
        QuickSort(arr, low, pi-1);
        printf("Sub array 2: (de %d): ", arr[pi]);
        printSubarray(arr, pi+1, high);
        QuickSort(arr, pi+1, high);
    }
}

#endif
```

```
Pivote: 225
Sub array 1 (de 225): 1 29 6 3 99 3 44 2 7
Pivote: 7
Sub array 1 (de 7): 1 6 3 3 2
Pivote: 2
Sub array 1 (de 2): 1
Sub array 2: (de 2): 3 3 6
Pivote: 6
Sub array 1 (de 6): 3 3
Pivote: 3
Sub array 1 (de 3): 3
Sub array 2: (de 3): 3
Sub array 2: (de 6): 44 99 29
Pivote: 29
Sub array 1 (de 29): 1 29 6 3 99 3 44 2 7
Sub array 2: (de 29): 99 44
Pivote: 44
Sub array 1 (de 44): 1 29 6 3 99 3 44 2 7
Sub array 2: (de 44): 99
Sub array 2: (de 225): 1 29 6 3 99 3 44 2 7
Arreglo ordenado:
1 2 3 3 6 7 29 44 99 225
-----
Process exited after 0.1583 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejecución de BubbleSort:

Se imprime en pantalla cada iteración necesaria para acomodar el arreglo y finalmente el arreglo ordenado. Cabe destacar que para que se compile y realice esta acción es necesario que se comente la función de QuickSort en el programa principal y descomentar la función bubbleSort.

```
#ifndef BubbleSort_h
#define BubbleSort_h
#include "fGenerics.h"

void bubbleSort (int a[], int size){
    int i, j;
    int index;
    i = size-1;
    while (i > 0){
        index = 0;
        for (j=0; j<i; j++){
            if (a[j+1] < a[j]){
                swap(&a[j+1], &a[j]);
                index = j;
            }
        }
        printArray(a, size);
        printf("Fin Iteracion \n");
        i = index;
    }
}

#endif
```

```
1 29 6 3 99 3 44 2 7 225
1 6 29 3 99 3 44 2 7 225
1 6 3 29 99 3 44 2 7 225
1 6 3 29 99 3 44 2 7 225
1 6 3 29 3 99 44 2 7 225
1 6 3 29 3 44 99 2 7 225
1 6 3 29 3 44 2 99 7 225
1 6 3 29 3 44 2 7 99 225
1 6 3 29 3 44 2 7 99 225
Fin Iteracion
1 3 6 3 29 2 7 44 99 225
1 3 6 3 29 2 7 44 99 225
1 3 6 29 2 7 44 99 225
1 3 6 29 2 7 44 99 225
1 3 6 2 29 7 44 99 225
1 3 6 2 29 7 44 99 225
Fin Iteracion
1 3 3 6 2 7 29 44 99 225
1 3 3 6 2 7 29 44 99 225
1 3 3 6 2 7 29 44 99 225
1 3 3 2 6 7 29 44 99 225
1 3 3 2 6 7 29 44 99 225
Fin Iteracion
1 3 3 2 6 7 29 44 99 225
1 3 3 2 6 7 29 44 99 225
1 3 2 3 6 7 29 44 99 225
1 2 3 3 6 7 29 44 99 225
Fin Iteracion
1 2 3 3 6 7 29 44 99 225
Arreglo ordenado:
1 2 3 3 6 7 29 44 99 225
```


The screenshot shows the Dev-C++ IDE with the file `Programa-principal.c` open. The code includes `<stdio.h>`, `"BubbleSort.h"`, and `"QuickSort.h"`. The `main` function initializes an array `arr` with the values `{1, 29, 6, 3, 99, 3, 44, 2, 7, 225}`, calculates its size, and calls `QuickSort` and `bubbleSort`. It then prints the sorted array and returns 0. The compiler window shows a warning: `[Warning] extra tokens at end of #ifndef directive [enabled by default]` and another: `[Warning] missing whitespace after the macro name [enabled by default]`.

```
1 #include <stdio.h>
2 #include "BubbleSort.h"
3 #include "QuickSort.h"
4
5 int main(){
6     int arr[] = {1,29,6,3,99,3,44,2,7,225};
7     int size = sizeof(arr)/sizeof(arr[0]);
8     //QuickSort (arr, 0, size-1);
9     bubbleSort(arr,size);
10    printf("Arreglo ordenado: \n");
11    printArray(arr,size);
12    return 0;
13 }
```

1.2.- Realizar una breve descripción de cada una de las funciones que tiene el código:

*Realizaré las breves descripciones de cada función dentro del código del mismo

Función principal:

The screenshot shows the Dev-C++ IDE with the file `Programa-principal.c` open. The code includes `<stdio.h>`, `"BubbleSort.h"`, and `"QuickSort.h"`. The `main` function initializes an array `arr` with the values `{1, 29, 6, 3, 99, 3, 44, 2, 7, 225}`, calculates its size, and calls `QuickSort` and `bubbleSort`. It then prints the sorted array and returns 0. The compiler window shows a warning: `[Warning] extra tokens at end of #ifndef directive [enabled by default]` and another: `[Warning] missing whitespace after the macro name [enabled by default]`.

```
1 /*En la primera version me equivoque al llamar los archivos
2 y ciertos detallitos que ya corrigi*/
3 #include <stdio.h>
4 #include "BubbleSort.h"
5 #include "QuickSort.h"
6
7 int main(){
8     int arr[] = {1,29,6,3,99,3,44,2,7,225};
9     int size = sizeof(arr)/sizeof(arr[0]);
10    //Se verifica el tamaño del arreglo
11
12    //QuickSort (arr, 0, size-1);
13    /*Se manda a llamar la funcion QuickSort de su archivo y se manda el arreglo,
14    un cero y el tamaño del arreglo-1 (9)*/
15
16    bubbleSort(arr,size);
17    /*Se manda a llamar la funcion bubbleSort de su archivo y se manda el arreglo
18    asi como su tamaño*/
19
20    printf("Arreglo ordenado: \n");
21    printArray(arr,size); //Se manda a imprimir el arreglo ordenado
22    return 0;
23 }
```

Archivo f.Genéricas:

```
C:\Users\israel\Desktop\P1Tita\Genericas.h - Dev-C++ 5.3.0.4
File Edit Search View Project Execute Debug Tools CVS Window Help

(globals)
Project Classes Debug [*]fGenericas.h [*]BubbleSort.h [*]QuickSort.h [*]Programa-principal.c purebavd.cpp

1 #ifndef _fGenericas.h_
2 #define _fGenericas.h_
3
4 void swap(int *a, int *b){
5     /*En esta funcion se realiza los intercambios del contenido variables cuando se mandan a
6     llamar, apoyandose de una variable temporal para no perder el valor de a, asi como
7     de los apuntadores para cambiar el contenido de las variables */
8     int t = *a;
9     *a = *b;
10    *b = t;
11 }
12
13 void printArray(int arr[],int size){
14     /*imprime el arreglo cuando se manda a llamar la funcion, apoyandose de un ciclo for*/
15     int i;
16     for (i=0; i < size; i++)
17         printf("%d ", arr[i]);
18     printf("\n");
19 }
20 #endif

Compiler Resources Compile Log Debug Find Results
Line: 20 Col: 7 Sel: 0 Lines: 20 Length: 596 Insert Modified
Escribe aquí para buscar
```

Archivo QuickSort.h :

```
C:\Users\israel\Desktop\P1Tita\QuickSort.h - [Executing] - Dev-C++ 5.3.0.4
File Edit Search View Project Execute Debug Tools CVS Window Help

(globals)
Project Classes Debug [*]fGenericas.h [*]BubbleSort.h [*]QuickSort.h Programa-principal.c purebavd.cpp

1 #ifndef _QuickSort.h_
2 #define _QuickSort.h_
3 #include "fGenericas.h"
4
5 void printSubArray (int arr[], int low, int high){
6     int i;
7     for (i=low;i<=high;i++)
8         printf("%d ", arr[i]);
9     printf("\n");
10    /*Funcion para imprimir los subarreglos que le manden*/
11 }
12
13 int partition (int arr[],int low, int high){
14     int pivot = arr[high];
15     /*Se define al pivote como la ultima posicion del arreglo
16     printf("Pivote: %d \n ", pivot);
17     int j,i = (low - 1);
18     for (j = low; j <= high - 1; j++){
19         if (arr[j] <= pivot){
20             i++;
21             swap(&arr[i],&arr[j]);
22             /*De ser menor o igual el valor de la variable a comparar con
23             la del pivote, realizar el intercambio de posicion*/
24         }
25     }
26     swap(&arr[i+1], &arr[high]);
27     return (i+1);
28 }
29
30 void QuickSort(int arr[],int low, int high){
31     /*Dentro de esta funcion se mandan a llamar las necesarias para realizar el ordenamiento, recibe
32     el arreglo, un cero y el tamaño del arreglo-1 (9)*/
33     if(low < high){ //si hay mas de un elemeto
34         int pi = partition(arr, low, high);
35         // se manda a llamar la funcion partition, mandando los parametros recibidos en QuickSort
36         printf("Sub array 1 (de %d): \t ",arr[pi]);
37         printSubArray(arr,low,pi-1);
38         QuickSort(arr,low,pi-1);
39         printf("Sub array 2: (de %d): \t ",arr[pi]);
40         printSubArray(arr,pi+1,high);
41     }
42 }
```

```

41         QuickSort(arr, pi + 1, high);
42         /*Se manda a imprimir el subarreglo uno y dos con la posicion del ultimo indice uno menor al pivote
43         respectivamente, de igual manera se vuelve a realizar la funcion QuickSort en estos subarreglos, para poder
44         ordenar todos los elementos del mismo*/
45     }
46 }
47 #endif
48

```

Compiler | Resources | Compile Log | Debug | Find Results | Close

Line	Col	File	Message
4	0	C:\Users\israel\Desktop\P1Tista\Programa-principal.c	In file included from C:\Users\israel\Desktop\P1Tista\Programa-principal.c
1	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] extra tokens at end of #ifndef directive [enabled by default]
2	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] missing whitespace after the macro name [enabled by default]

Line: 48 Col: 1 Sel: 0 Lines: 48 Length: 1725 Insert Done parsing

Escribe aquí para buscar

05:20 p. m. 18/08/2018

Archivo BubbleSort.h:

C:\Users\israel\Desktop\P1Tista\BubbleSort.h - [Executing] - Dev-C++ 5.3.0.4

File Edit Search View Project Execute Debug Tools CVS Window Help

(globals)

Project Classes Debug [*] fGenericas.h [*] BubbleSort.h [*] QuickSort.h Programa-principal.c purebavd.cpp

```

1 #ifndef _BubbleSort.h_
2 #define _BubbleSort.h_
3 #include "fGenericas.h"
4 void bubbleSort(int a[], int size){
5     /*recibe el arreglo y el tamaño del mismo de la funcion principal*/
6     int i,j;
7     int index;
8     i= size-1;
9     while (i>0){
10         index = 0;
11         for(j=0;j<i;j++){
12             if(a[j+1]<a[j]){
13                 swap(&a[j+1],&a[j]);
14                 index=j;
15             }
16             /*Se compara al la variable que va delante de la que se selecciona con el valor
17             del indice de j, en caso de que esta sea menor, se intercambia los valores y se le
18             asigna al indice el valor actual de j, para de esta manera no se repitan las
19             comparaciones */
20             printArray(a,size); //se manda a imprimir el arreglo
21         }
22         printf("Fin Iteracion \n");
23         i=index; // se le asigna a i el tamaño del ultimo indice comparado
24     }
25 }
26 #endif

```

Compiler | Resources | Compile Log | Debug | Find Results | Close

Line	Col	File	Message
4	0	C:\Users\israel\Desktop\P1Tista\Programa-principal.c	In file included from C:\Users\israel\Desktop\P1Tista\Programa-principal.c
1	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] extra tokens at end of #ifndef directive [enabled by default]
2	20	C:\Users\israel\Desktop\P1Tista\BubbleSort.h	[Warning] missing whitespace after the macro name [enabled by default]

Line: 26 Col: 7 Sel: 0 Lines: 26 Length: 873 Insert Modified

Escribe aquí para buscar

05:10 p. m. 18/08/2018

1.3.- Indicar las similitudes y diferencias que tiene la implementaciones con respecto a las que se vieron en clase:

- QuickSort: Entre sus similitudes está el procedimiento en sí, donde se usa un pivote para ser comparado con los demás elementos. Por otro lado en sus diferencias, se toma como pivote al último elemento del arreglo, que en comparación a lo visto en clase, se usa el primero de manera recomendada,

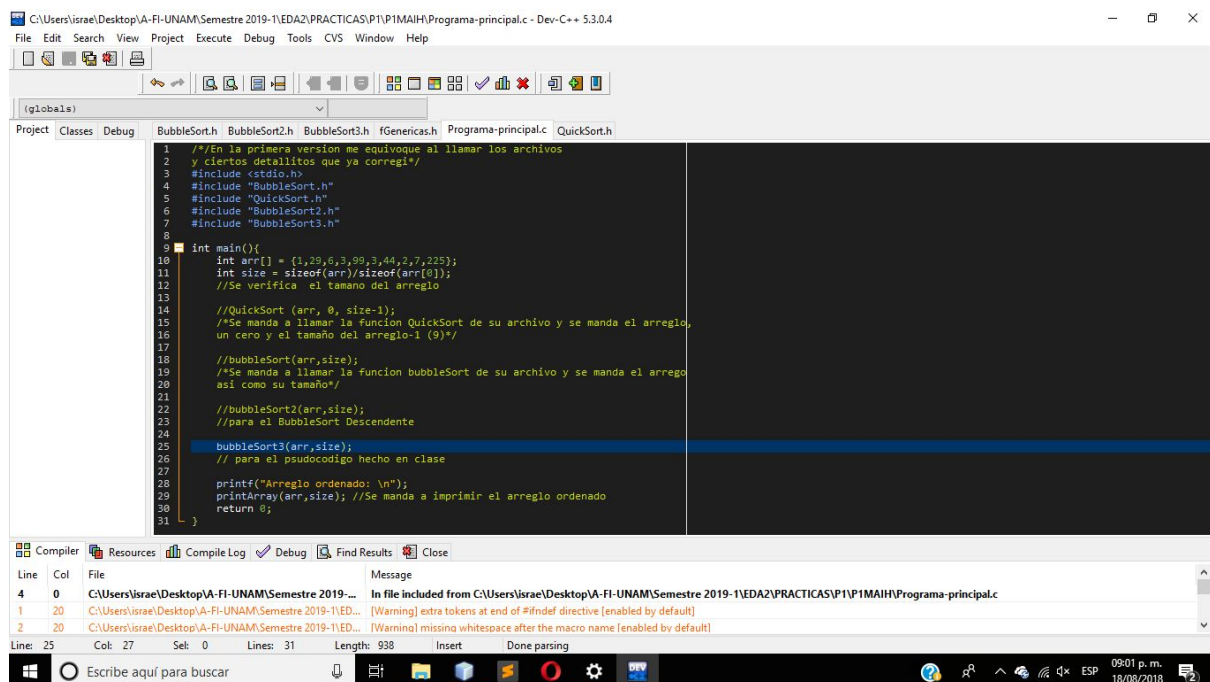
de igual manera en el código de la práctica se imprime a pantalla las dos sublistas en cada iteración.

- BubbleSort: El algoritmo usa los mismos pasos, sin embargo, en el código de la práctica se hace uso de un índice y se comparan los elementos de derecha a izquierda.

1.4.-Realiza la implementación de Bubble Sort basada en el pseudocódigo visto en clase. Realiza los comentarios respectivos: modifique la función principal para que reconociera a la versión de Bubble Sort visto en clase.

Para ello cree el archivo BubbleSort3.h, en el cual reutilice el pseudocódigo visto en clase para la pasarlo a lenguaje C y usando la función swap y printArray en el orden correcto, ahora si funciona el código.

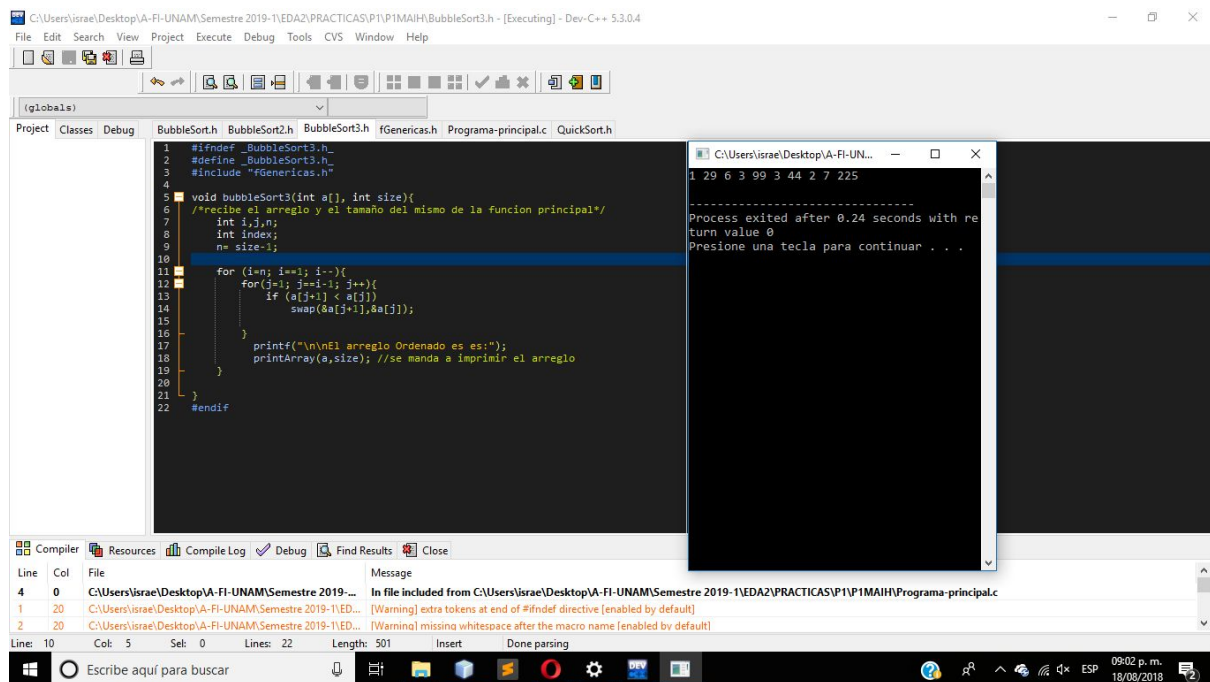
Usare dos capturas de pantallas para esto, el primero es para que se vea modificaciones a la función principal y la que sigue es del archivo BubbleSort3.h con la captura de la consola con los números ya ordenados.



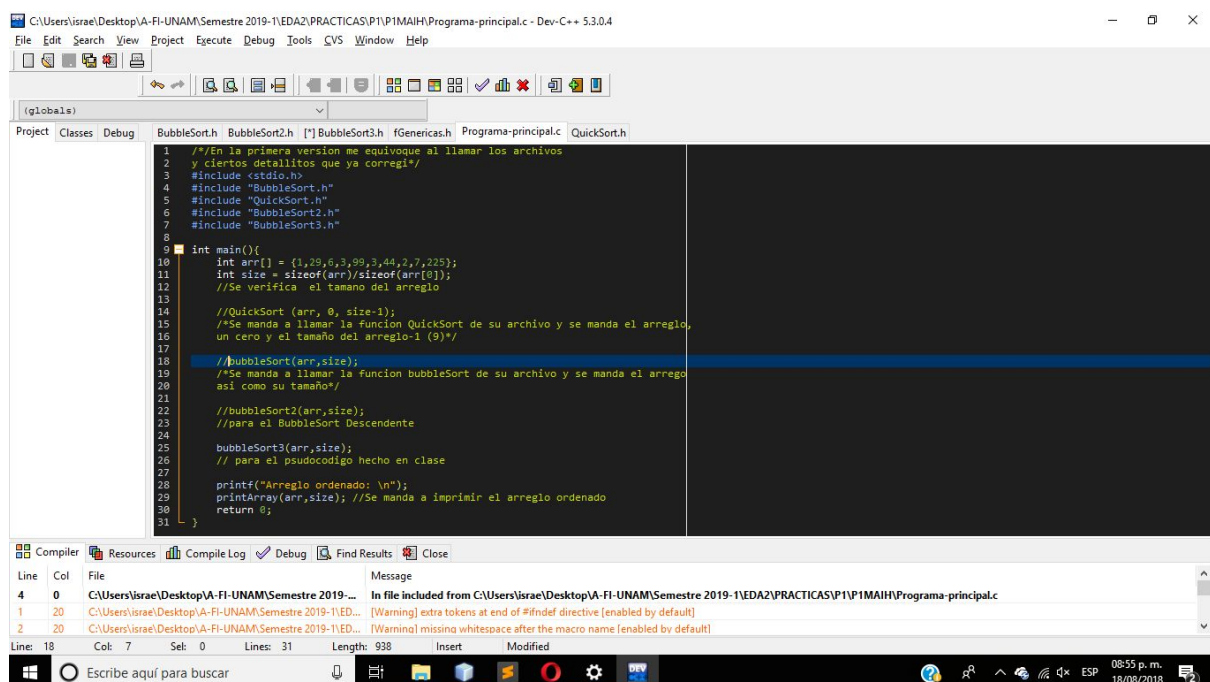
```
1 //En la primera versión me equivoque al llamar los archivos
2 y ciertos detallitos que ya corregí
3 #include <stdio.h>
4 #include "BubbleSort.h"
5 #include "QuickSort.h"
6 #include "BubbleSort2.h"
7 #include "BubbleSort3.h"
8
9 int main()
10 {
11     int arr[] = {1,2,3,4,5,6,7,8,9,10};
12     int size = sizeof(arr)/sizeof(arr[0]);
13     //Se verifica el tamaño del arreglo
14     //QuickSort (arr, 0, size-1);
15     //Se manda a llamar la función QuickSort de su archivo y se manda el arreglo,
16     //un cero y el tamaño del arreglo-1 (9)
17     //bubbleSort(arr, size);
18     //Se manda a llamar la función bubbleSort de su archivo y se manda el arreglo
19     //así como su tamaño
20     //bubbleSort2(arr, size);
21     //para el BubbleSort Descendente
22     bubbleSort3(arr, size);
23     // para el pseudocódigo hecho en clase
24
25     printf("Arreglo ordenado: \n");
26     printArray(arr, size); //Se manda a imprimir el arreglo ordenado
27     return 0;
28 }
```

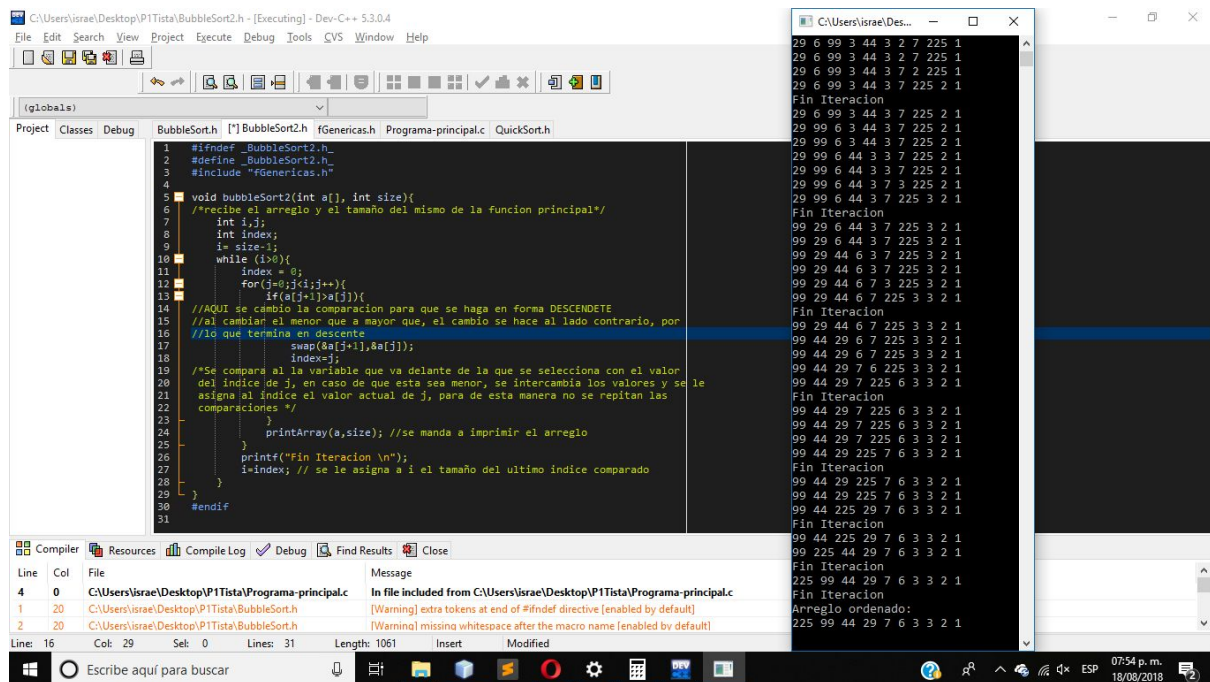
Line: 25 Col: 27 Sel: 0 Lines: 31 Length: 938 Insert Done parsing

Message
In file included from C:\Users\israel\Desktop\A-FI-UNAM\Semestre 2019-1\EDAA2\PRACTICAS\P1\P1MAIH\Programa-principal.c
[Warning] extra tokens at end of #ifndef directive [enabled by default]
[Warning] missing whitespace after the macro name [enabled by default]



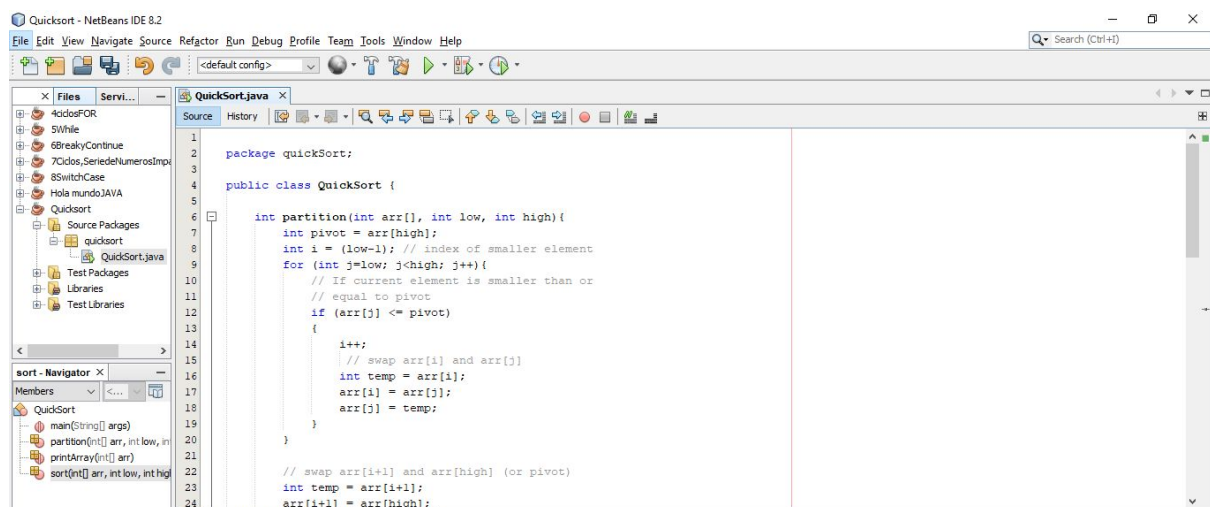
1.5.- Indica las modificaciones que se deben realizar para que el ordenamiento sea descendente. (indica las funciones modificadas y una captura de pantalla con los resultados):

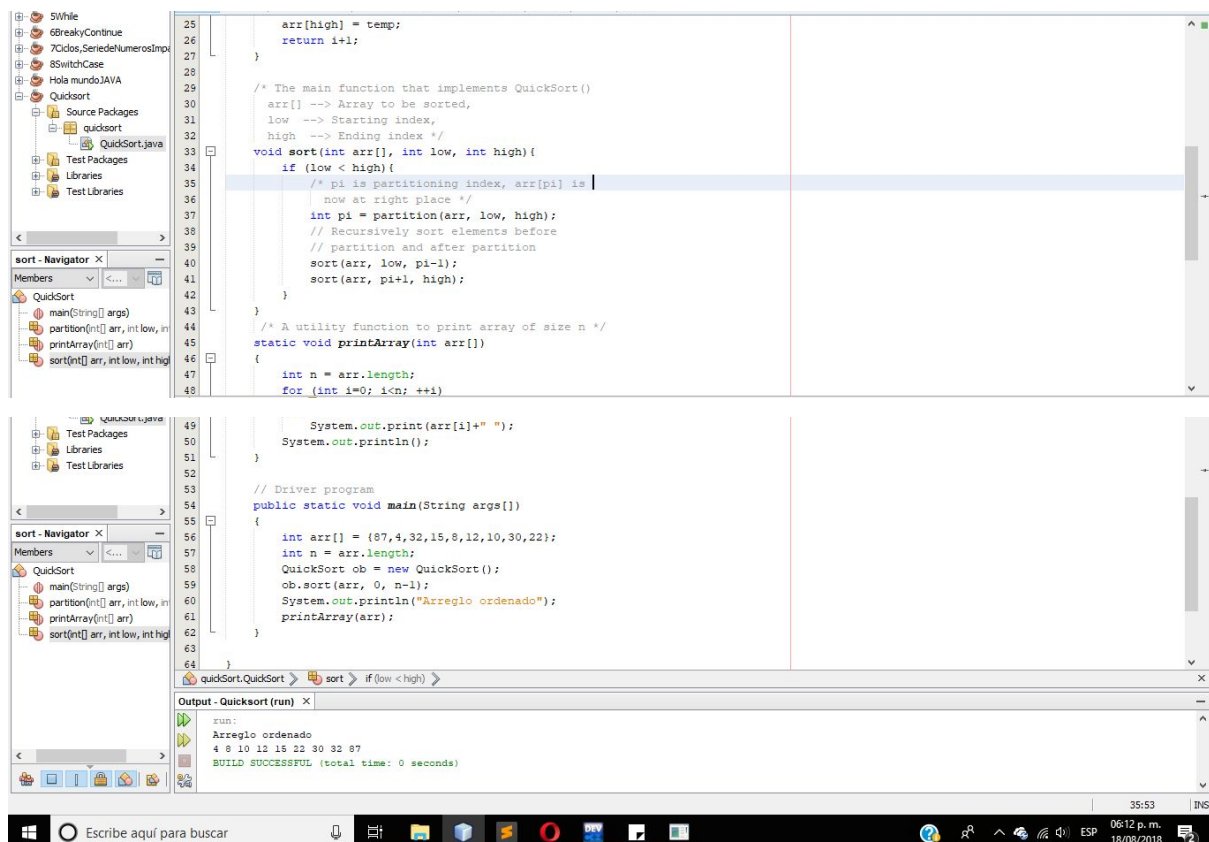




modifique el código dado en la práctica creando el archivo BubbleSort2.h, donde al cambiar el menor que por un mayor que en la comparación y de esta manera al realizar el cambio con la función swap, el sentido se invierte siendo descendente.

2.- QuickSort en JAVA: Abre el proyecto “QuickSort”, compila y ejecuta el proyecto. Revisa el código y realiza comentarios de los aspectos que te llamen la atención en JAVA.





De lo que me llamó la atención del algoritmo de QuickSort en JAVA fue el gran parecido que tiene con el lenguaje c, sin embargo, tiene cosas muy distintas, como la manera de imprimir a pantalla, la de declarar arreglos, obtener el tamaño de los mismos y de que para este proyecto se de más archivos si lo podemos llamar así, ya que en el zip venían demasiadas carpetas para un solo programa, donde en C, solo seria un archivo, si acomodamos todo en un solo código como venía este.

Conclusiones:

Se cumplió el objetivo en el que ahora identifico mejor los algoritmos QuickSort y BubbleSort. Ahora tengo no solo la idea de en qué consiste cada algoritmo, sino que también, ya comprendo al código necesario para realizarlos.

Cabe destacar que esta práctica no solo me ayudo a poner en código los temas vistos en clase, sino que igual aprendí como usar distintos archivos dentro del código y por otro lado, pude aprender un poco de JAVA gracias a la última actividad de la práctica.