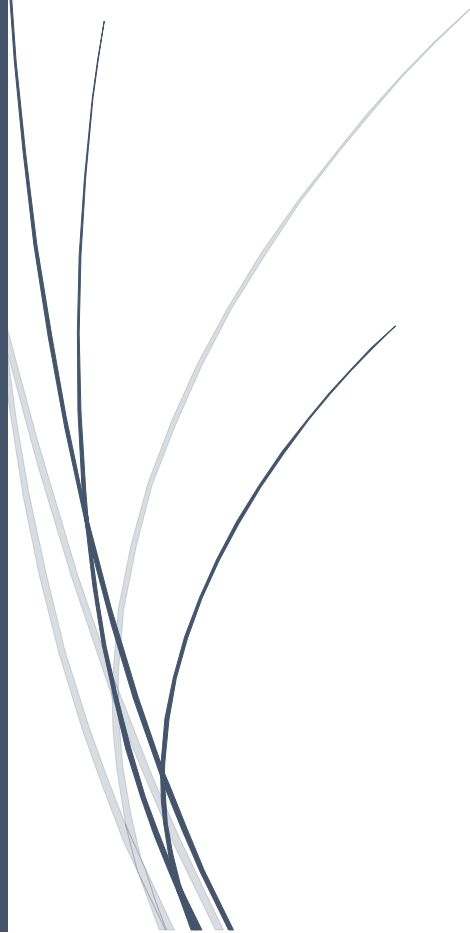


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

10/19/2021

Números complejos

Practica Tema 4

Several thin, curved, wavy lines in dark blue and light grey on the left side of the page.

Alejandro Colmenero Moreno
ASIGNATURA

ÍNDICE

ENUNCIADO.....	2
CÓDIGO	3
Main.c.....	3
Complejos.h.....	4
Complejos.c	5
EJECUCIÓN.....	6
ERRORES.....	7
DOCUMENTACIÓN GENERADA.....	8
BIBLIOGRAFÍA	10

ENUNCIADO

1. Se debe modularizar en tres ficheros:

- a. main.c
- b. complejos.h
- c. complejos.c

2. Debemos crear una estructura llamada `numeroComplejo` compuesta por dos números enteros, uno llamada real y el otro imaginario.

3. En la librería `complejos.h` debemos declarar las siguientes funciones:

- a. `sumar()` que recibe dos números complejos y devuelve otro número complejo resultado de la suma de los pasados como argumentos
- b. `restar()`, ídem pero restando
- c. `multiplicar()`, ídem pero multiplicando

4. En el fichero `complejos.c` debemos implementar las funciones antes declaradas.

5. En el programa principal debemos pedir los valores de dos números complejos por teclado tras lo cual enviaremos a las funciones antes descritas y mostraremos por pantalla el resultado de sumar, restar y multiplicar los dos números complejos creados.

6. Se debe crear, en formato HTML, la documentación de todo el proyecto; usad variables globales para poder incluirlas en dicha documentación. Usad Doxygen tal como se ha visto en el Temario.

CÓDIGO

Main.c

```
/** @file main.c
 * @brief Programa principal
 */
#include <stdio.h>
#include "complejos.h"

typedef struct numeroComplejo numeroComplejo;

/** @fn int main()
 * @brief Programa principal, pide los datos de 2 numeros complejos y los suma, resta y
multiplica.
 * Declaramos los 3 numeros complejos
 * Usamos las funciones suma(), resta(), multiplicacion()
 */
/** @var struct numeroComplejo nc1, nc2, nc3;
 * @brief numeros complejos con los que operamos
 */
int main()
{
    struct numeroComplejo nc1, nc2, nc3;

    fill_complejo(&nc1);
    fill_complejo(&nc2);

    nc3 = sumar(&nc1, &nc2);
    printf("Suma: %f, %f\n", nc3.real, nc3.imaginario );

    nc3 = restar(&nc1, &nc2);
    printf("Resta: %f, %f\n", nc3.real, nc3.imaginario );

    nc3 = multiplicar(&nc1, &nc2);
    printf("Multiplicacion: %f, %f\n", nc3.real, nc3.imaginario );

    return 0;
}
```

Complejos.h

```
/** @fn struct numeroComplejo sumar(numeroComplejo *nc1, numeroComplejo *nc2)
    @brief suma dos números complejos

    @param *nc1 Primer número complejo
    @param *nc1 Segundo número complejo
*/
/** @fn struct numeroComplejo restar(numeroComplejo *nc1, numeroComplejo *nc2)
    @brief resta dos números complejos

    @param *nc1 Primer número complejo
    @param *nc1 Segundo número complejo
*/
/** @fn struct numeroComplejo multiplicar(numeroComplejo *nc1, numeroComplejo *nc2)
    @brief multiplica dos números complejos

    @param *nc1 Primer número complejo
    @param *nc1 Segundo número complejo
*/
/** @fn void fill_complejo(struct numeroComplejo *nc)
    @brief Pide por pantalla el número real e imaginario.

    @param *nc1 Número complejo que rellenar
*/
typedef struct numeroComplejo numeroComplejo;
typedef struct numeroComplejo{
    float real;
    float imaginario;
};

void fill_complejo(numeroComplejo *nc);
struct numeroComplejo sumar(numeroComplejo *nc1, numeroComplejo *nc2);
struct numeroComplejo restar(numeroComplejo *nc1, numeroComplejo *nc2);
struct numeroComplejo multiplicar(numeroComplejo *nc1, numeroComplejo *nc2);
```

Complejos.c

```
typedef struct numeroComplejo numeroComplejo;
typedef struct numeroComplejo{
    float real;
    float imaginario;
};

struct numeroComplejo sumar(numeroComplejo *nc1, numeroComplejo *nc2){
    struct numeroComplejo nc3;
    nc3.real = nc1->real + nc2->real;
    nc3.imaginario = nc1->imaginario + nc2->imaginario;
    return nc3;
}

struct numeroComplejo restar(numeroComplejo *nc1, numeroComplejo *nc2){
    struct numeroComplejo nc3;
    nc3.real = nc1->real - nc2->real;
    nc3.imaginario = nc1->imaginario - nc2->imaginario;
    return nc3;
}

struct numeroComplejo multiplicar(numeroComplejo *nc1, numeroComplejo *nc2){
    struct numeroComplejo nc3;
    nc3.real = nc1->real * nc2->real;
    nc3.imaginario = nc1->imaginario * nc2->imaginario;
    return nc3;
}

void fill_complejo(struct numeroComplejo *nc){

    float real = 0, imaginario = 0;

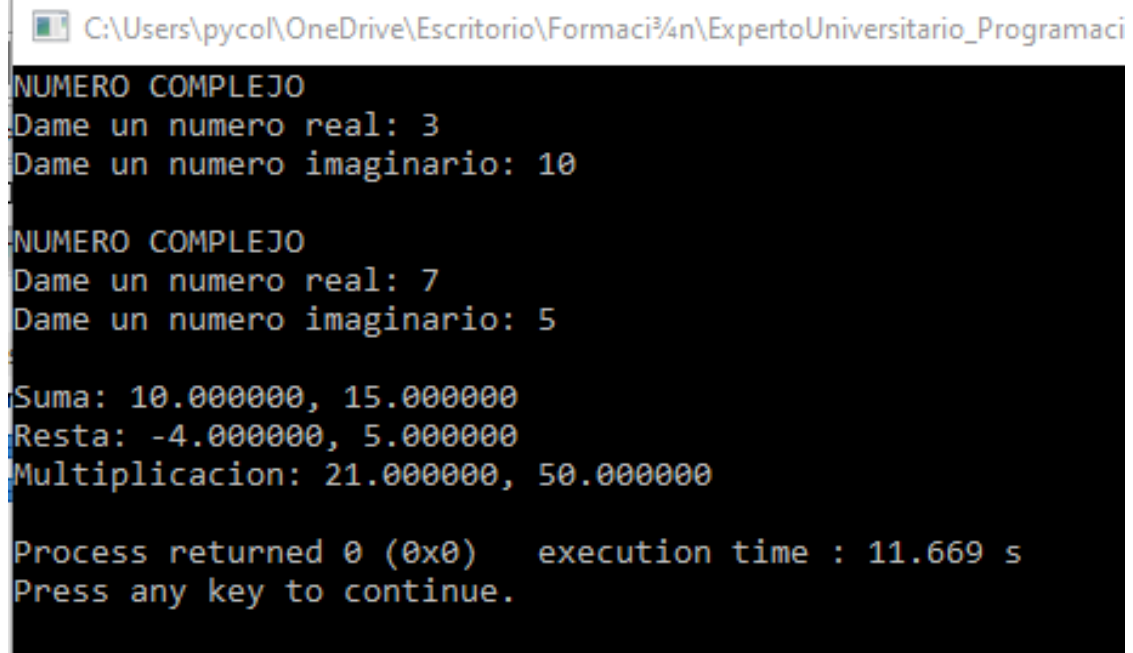
    printf("NUMERO COMPLEJO \n");
    printf("Dame un numero real: ");
    scanf("%f", &real);
    printf("Dame un numero imaginario: ");
    scanf("%f", &imaginario);

    nc->real = real;
    nc->imaginario = imaginario;

    printf("\n");
}
```

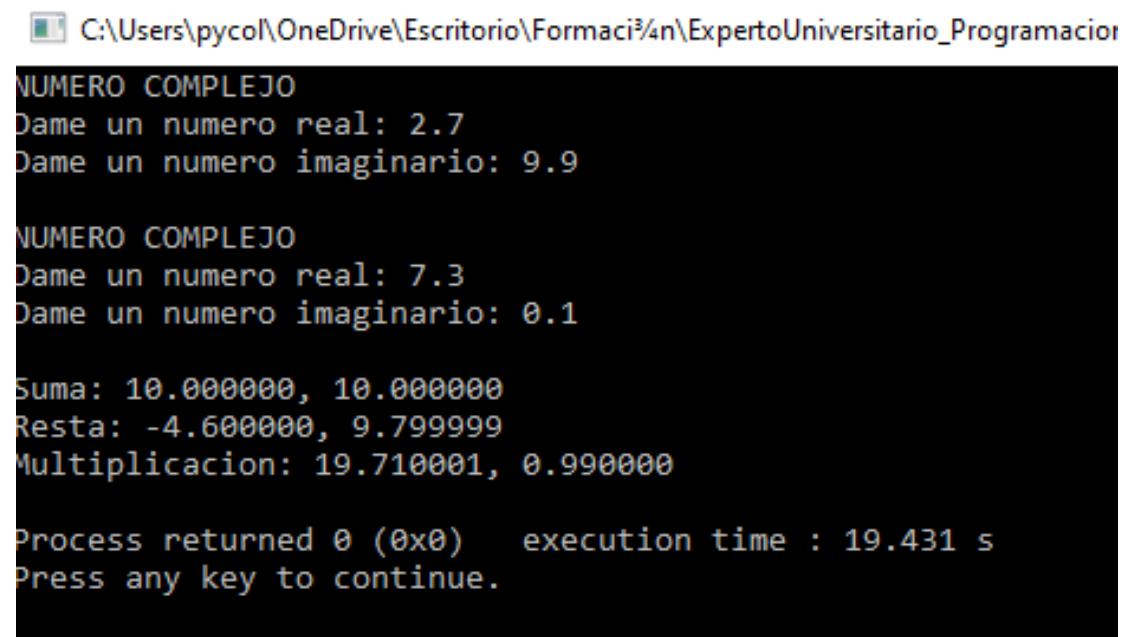
EJECUCIÓN

Primer ejemplo, números enteros:



```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Programaci  
NUMERO COMPLEJO  
Dame un numero real: 3  
Dame un numero imaginario: 10  
  
NUMERO COMPLEJO  
Dame un numero real: 7  
Dame un numero imaginario: 5  
  
Suma: 10.000000, 15.000000  
Resta: -4.000000, 5.000000  
Multiplicacion: 21.000000, 50.000000  
  
Process returned 0 (0x0)   execution time : 11.669 s  
Press any key to continue.
```

Segundo ejemplo, números reales:



```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Programaci  
NUMERO COMPLEJO  
Dame un numero real: 2.7  
Dame un numero imaginario: 9.9  
  
NUMERO COMPLEJO  
Dame un numero real: 7.3  
Dame un numero imaginario: 0.1  
  
Suma: 10.000000, 10.000000  
Resta: -4.600000, 9.799999  
Multiplicacion: 19.710001, 0.990000  
  
Process returned 0 (0x0)   execution time : 19.431 s  
Press any key to continue.
```

ERRORES

Si pongo letras en alguna pedida de números, lo intenta convertir a Float, si no puede lo deja en 0.0 y si hago salto de líneas, hasta que no escriba algo no continua la ejecución del código.

 C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Programacion\

```
NUMERO COMPLEJO
Dame un numero real: 3a
Dame un numero imaginario: 7

NUMERO COMPLEJO
Dame un numero real: 10
Dame un numero imaginario:

h

Suma: 13.000000, 7.000000
Resta: -7.000000, 7.000000
Multiplicacion: 30.000000, 0.000000

Process returned 0 (0x0)   execution time : 11.757 s
Press any key to continue.
```


DOCUMENTACIÓN GENERADA

Genero el archivo de configuración

```
C:\Users\pycol\OneDrive\Escritorio\Formación\ExpertoUniversitario_ProgramacionAvanzada\FundamentosProgramacion2\Tema4\PracticaT4>doxygen -g config.cfg

Configuration file 'config.cfg' created.

Now edit the configuration file and enter

doxygen config.cfg

to generate the documentation for your project
```

Lo edito:

```
34
35 PROJECT_NAME           = "Practica Programacion2 T4"
36
```

```
93
94 OUTPUT_LANGUAGE       = Spanish
95
```

```
271
272 OPTIMIZE_OUTPUT_FOR_C = YES
273
```

Genero la documentación:

```
Generating docs for compound numeroComplejo...
C:/Users/pycol/Desktop/LoremIpsum/ke/complejos.c:6: warning: Member re
C:/Users/pycol/Desktop/LoremIpsum/ke/complejos.c:7: warning: Member im
Generating concept documentation...
Generating namespace index...
Generating graph info page...
Generating directory documentation...
Generating index page...
Generating page index...
Generating module index...
Generating namespace index...
Generating namespace member index...
Generating concept index...
Generating annotated compound index...
Generating alphabetical compound index...
Generating hierarchical class index...
Generating member index...
Generating file index...
Generating file member index...
Generating example index...
finalizing index lists...
writing tag file...
Running plantuml with JAVA...
lookup cache used 7/65536 hits=81 misses=8
finished...
```

Practica Programacion2 T4

Página principal	Estructuras de Datos ▾	Archivos ▾	<input type="text" value="Q* Buscar"/>
----------------------------------	--	----------------------------	--

Campos de datos

Referencia de la Estructura numeroComplejo

Campos de datos

float	real
float	imaginario

La documentación para esta estructura fue generada a partir de los siguientes ficheros:

- [complejos.c](#)
- [complejos.h](#)

BIBLIOGRAFÍA

<https://es.stackoverflow.com/questions/309124/c-error-is-a-pointer-did-you-mean-to-use>

<https://stackoverflow.com/questions/36217352/unknown-type-name-error-in-c>

<https://stackoverflow.com/questions/60064157/error-incompatible-type-for-argument-1-of-push>