



C Piscine

C 05

*Resumen: Este documento corresponde al enunciado del módulo C 05 de la C Piscine de 42.*

*Versión: 6.2*

# Índice general

|       |                                       |    |
|-------|---------------------------------------|----|
| I.    | Instrucciones                         | 2  |
| II.   | Introducción                          | 4  |
| III.  | Ejercicio 00 : ft_iterative_factorial | 6  |
| IV.   | Ejercicio 01 : ft_recursive_factorial | 7  |
| V.    | Ejercicio 02 : ft_iterative_power     | 8  |
| VI.   | Ejercicio 03 : ft_recursive_power     | 9  |
| VII.  | Ejercicio 04 : ft_fibonacci           | 10 |
| VIII. | Ejercicio 05 : ft_sqrt                | 11 |
| IX.   | Ejercicio 06 : ft_is_prime            | 12 |
| X.    | Ejercicio 07 : ft_find_next_prime     | 13 |
| XI.   | Ejercicio 08 : Las diez damas         | 14 |
| XII.  | Entrega y evaluación                  | 15 |

# Capítulo I

## Instrucciones

- Esta página será la única referencia: no te fíes de los rumores.
- ¡Ten cuidado! Los enunciados pueden cambiar en cualquier momento.
- Asegúrate de que tus directorios y archivos tienen los permisos adecuados.
- Debes respetar el procedimiento de entrega para todos tus ejercicios.
- Tus compañeros de piscina se encargarán de corregir tus ejercicios.
- Además de por tus compañeros, también serán corregidos por un programa que se llama la Moulinette.
- La Moulinette es muy estricta a la hora de evaluar. Está completamente automatizada. Es imposible discutir con ella sobre tu nota. Por lo tanto, se extremadamente riguroso para evitar cualquier sorpresa.
- La Moulinette no tiene una mente muy abierta. No intenta comprender el código que no respeta la Norma. La Moulinette utiliza el programa **norminette** para comprobar La Norma en sus archivos. Entiende entonces que es estúpido entregar un código que no pase la **norminette**.
- Los ejercicios han sido ordenados con mucha precisión, del más sencillo al más complejo. En ningún caso se tendrá en cuenta un ejercicio complejo si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- El uso de una función prohibida se considera una trampa. Cualquier trampa será sancionada con la nota -42.
- Solamente hay que entregar una función `main()` si lo que se pide es un programa.
- La Moulinette compila con los flags `-Wall -Wextra -Werror` y utiliza `cc`.
- Si tu programa no compila, tendrán un 0.
- No puedes dejar en tu directorio ningún archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.
- ¿Tienes alguna pregunta? Pregunta a tu compañero de la derecha. Si no, prueba con tu compañero de la izquierda.

- Tu manual de referencia se llama `Google / man / Internet / ....`
- ¡No olvides participar en el slack de tu Piscina!
- Lee detenidamente los ejemplos. Podrían exigir cosas que no se especifican necesariamente en los enunciados...
- Razona. ¡Te lo suplico, por Thor, por Odín! Maldita sea.



Para este módulo, la Norminette debe ser ejecutada con el flag `-R CheckForbiddenSourceHeader`. La Moulinette también lo utilizará.

# Capítulo II

## Introducción

He aquí un texto extraído del primer libro de la saga de Harry Potter:

Oh, podrás pensar que no soy bonito,  
pero no juzgues por lo que ves.  
Me comeré a mí mismo si puedes encontrar  
un sombrero más inteligente que yo.

Puedes tener bombines negros,  
sombreros altos y elegantes.  
Pero yo soy el Sombrero Seleccionador de Hogwarts  
y puedo superar a todos.

No hay nada escondido en tu cabeza  
que el Sombrero Seleccionador no pueda ver.  
Así que pruébame y te diré  
dónde debes estar.

Puedes pertenecer a Gryffindor,  
donde habitan los valientes.  
Su osadía, temple y caballerosidad  
ponen aparte a los de Gryffindor.

Puedes pertenecer a Hufflepuff  
donde son justos y leales.  
Esos perseverantes Hufflepuff  
de verdad no temen el trabajo pesado.

O tal vez a la antigua sabiduría de Ravenclaw,  
Si tienes una mente dispuesta,  
porque los de inteligencia y erudición  
siempre encontrarán allí a sus semejantes.


O tal vez en Slytherin  
harás tus verdaderos amigos.  
Esa gente astuta utiliza cualquier medio  
para lograr sus fines.

¡Así que pruébame! ¡No tengas miedo!  
¡Y no recibirás una bofetada!  
Estás en buenas manos (aunque yo no las tenga).  
Porque soy el Sombrero Pensante.

Desgraciadamente, este tema no tiene nada que ver con la saga de Harry Potter y es una lástima, porque el proyecto no se va a realizar por arte de magia.

# Capítulo III

## Ejercicio 00 : ft\_iterative\_factorial


|   |               |
|---|---------------|
|  | Ejercicio: 00 |
| ft_iterative_factorial  |               |
| Directorio de entrega: <i>ex00/</i>   |               |
| Archivos a entregar: <b>ft_iterative_factorial.c</b>                              |               |
| Funciones autorizadas: Ninguna  |               |

- Escribe una función iterativa que devuelva un número. Este número será el resultado de la operación factorial a partir del número usado como parámetro.
- Si el argumento no es válido, la función debe devolver 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- El prototipo de la función deberá ser el siguiente:

```
int ft_iterative_factorial(int nb);
```

# Capítulo IV

## Ejercicio 01 : ft\_recursive\_factorial

|   |               |
|---|---------------|
|  | Ejercicio: 01 |
| ft_recursive_factorial  |               |
| Directorio de entrega: <i>ex01/</i>   |               |
| Archivos a entregar: <b>ft_recursive_factorial.c</b>                              |               |
| Funciones autorizadas: Ninguna  |               |


- Escribe una función recursiva que devuelva el factorial de un número usado como parámetro.
- Si el argumento no es válido, la función debe devolver 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- El prototipo de la función deberá ser el siguiente:

```
int ft_recursive_factorial(int nb);
```



# Capítulo V

## Ejercicio 02 : ft\_iterative\_power


|   |                    |
|---|--------------------|
|  | Ejercicio: 02      |
|   | ft_iterative_power |
| Directorio de entrega: <i>ex02/</i>   |                    |
| Archivos a entregar: <b>ft_iterative_power.c</b>                                  |                    |
| Funciones autorizadas: Ninguna  |                    |

- Escribe una función iterativa que devuelva una potencia de un número. Una potencia inferior a 0 devolverá 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- Se ha decidido que 0 potencia 0 devolverá 1
- El prototipo de la función deberá ser el siguiente:

```
int ft_iterative_power(int nb, int power);
```

# Capítulo VI

## Ejercicio 03 : ft\_recursive\_power


|   |  |
|---|--|
|  | Ejercicio: 03                                    |
|   | ft_recursive_power                               |
|   | Directorio de entrega: <i>ex03/</i>              |
|   | Archivos a entregar: <b>ft_recursive_power.c</b> |
|   | Funciones autorizadas: Ninguna                   |

- Escribe una función recursiva que devuelva una potencia de un número.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- Hemos decidido que 0 potencia 0 devolverá 1
- El prototipo de la función deberá ser el siguiente:

```
int ft_recursive_power(int nb, int power);
```

# Capítulo VII

## Ejercicio 04 : ft\_fibonacci

|   |  |
|---|--|
|  | Ejercicio: 04                              |
|   | ft_fibonacci                               |
|   | Directorio de entrega: <i>ex04/</i>        |
|   | Archivos a entregar: <b>ft_fibonacci.c</b> |
|   | Funciones autorizadas: <b>Ninguna</b>      |


- Escribe una función `ft_fibonacci` que devuelva el **n**-ésimo elemento de la sucesión de Fibonacci; El primer elemento de la serie está en el índice 0. Consideraremos que la sucesión de Fibonacci comienza por 0, 1, 1, 2.
- No se tendrán que gestionar los overflows.
- El prototipo de la función deberá ser el siguiente:

```
int ft_fibonacci(int index);
```

- Por supuesto, `ft_fibonacci` tendrá que ser recursiva.
- Si el `índice` es inferior a 0, la función retornará -1.

# Capítulo VIII

## Ejercicio 05 : ft\_sqrt


|   |               |
|---|---------------|
|  | Ejercicio: 05 |
|   | ft_sqrt       |
| Directorio de entrega: <i>ex05/</i>   |               |
| Archivos a entregar: <b>ft_sqrt.c</b>   |               |
| Funciones autorizadas: <b>Ninguna</b>   |               |

- Escribe una función que devuelva la raíz cuadrada entera de un número, si existe, o 0 si la raíz cuadrada no es entera.
- El prototipo de la función deberá ser el siguiente:

```
int ft_sqrt(int nb);
```

# Capítulo IX

## Ejercicio 06 : ft\_is\_prime

|   |               |
|---|---------------|
|  | Ejercicio: 06 |
|   | ft_is_prime   |
| Directorio de entrega: <i>ex06/</i>   |               |
| Archivos a entregar: <b>ft_is_prime.c</b>   |               |
| Funciones autorizadas: Ninguna  |               |

- Escribe una función que devuelva 1 si el número es primo y 0 si el número no lo es.
- El prototipo de la función deberá ser el siguiente:


```
int ft_is_prime(int nb);
```



0 y 1 no son números primos.

# Capítulo X

## Ejercicio 07 : ft\_find\_next\_prime


|   |               |
|---|---------------|
|  | Ejercicio: 07 |
| ft_find_next_prime  |               |
| Directorio de entrega: <i>ex07/</i>   |               |
| Archivos a entregar: <b>ft_find_next_prime.c</b>                                  |               |
| Funciones autorizadas: Ninguna  |               |

- Escribe una función que devuelva el número primo inmediatamente superior o igual al número usado como parámetro.
- El prototipo de la función deberá ser el siguiente:

```
int ft_find_next_prime(int nb);
```

# Capítulo XI

## Ejercicio 08 : Las diez damas

|   |               |
|---|---------------|
|  | Ejercicio: 08 |
| Las diez damas  |               |
| Directorio de entrega: <i>ex08/</i>   |               |
| Archivos a entregar: <b>ft_ten_queens_puzzle.c</b>                                |               |
| Funciones autorizadas: <b>write</b>   |               |

- Escribe una función que muestre todas las posibilidades de colocar diez damas en un tablero de 10x10 sin que se puedan alcanzar con una sola jugada y que devuelva el número de posibilidades.
- Se tendrá que utilizar la recursividad.
- El valor retornado por su función tendrá que ser el número de soluciones mostradas.
- El prototipo de la función deberá ser el siguiente:

```
int ft_ten_queens_puzzle(void);
```

- La visualización se hará de la manera siguiente:

```
$>./a.out | cat -e
0257948136$
0258693147$
...
4605713829$
4609582731$
...
9742051863$
$>
```

- La sucesión se lee de izquierda a derecha. La primera cifra corresponde a la posición de la primera dama en la primera columna (el índice empieza con 0). La n-ésima cifra corresponde a la posición de la n-ésima dama en la n-ésima columna.

# Capítulo XII

## Entrega y evaluación

Entrega tu proyecto en tu repositorio `Git` como de costumbre. Solo el trabajo entregado en el repositorio será evaluado durante la defensa. No dudes en comprobar varias veces los nombres de los archivos para verificar que sean correctos.



Sólo necesitas entregar los archivos requeridos por el enunciado de este proyecto.