



Piscina C

C 05

Resumen: Este documento corresponde a la evaluación del módulo C 05 de la piscina de 42.

Índice general

I.	Instrucciones	2
II.	Preámbulo	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

Capítulo I

Instrucciones

- Esta página será la única referencia: no se fíe de los rumores de pasillo.
- Vuelva a leer bien los enunciados antes de entregar sus ejercicios. Los enunciados pueden cambiar en cualquier momento.
- Tenga cuidado con los permisos de sus archivos y de sus directorios.
- Debe respetar el procedimiento de entrega para todos sus ejercicios.
- Sus compañeros de piscina se encargarán de corregir sus ejercicios.
- Además de por sus compañeros, también será corregido por un programa que se llama la Moulinette.
- La Moulinette es muy estricta a la hora de dar notas. Está completamente automatizada. Es imposible discutir con ella sobre su nota. Por lo tanto, sea 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 de sus archivos. Entienda 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 le prestaremos atención ni tendremos 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 tendrá que entregar una función `main()` si le pedimos un programa.
- La Moulinette compila con los flags `-Wall -Wextra -Werror` y utiliza `gcc`.
- Si su programa no compila, tendrá 0.
- No debe dejar en su directorio ningún archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.

- ¿Tiene alguna pregunta? Pregunte a su vecino de la derecha. Si no, pruebe con su vecino de la izquierda.
- Su manual de referencia se llama `Google / man / Internet /`
- ¡No olvide participar en el foro Piscina de su Intranet o en el slack de su Piscina!
- Lea detenidamente los ejemplos. Podrían exigir cosas que no se especifican necesariamente en los enunciados...
- Razone. ¡Se lo suplico, por Odín! Maldita sea.



Para esta jornada, la *norminette* debe ser ejecutada con el flag `-R CheckForbiddenSourceHeader`. La *moulinette* también lo utilizará.

Capítulo II

Preámbulo

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 la entrega 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>	
Ficheros a entregar : ft_iterative_factorial.c	
Funciones autorizadas : Ninguna	

- Escriba una función iterativa que devuelva un número. Este número será el resultado de la operación factorial a partir del número pasado 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>
	Ficheros a entregar : ft_recursive_factorial.c
	Funciones autorizadas : Ninguna

- Escriba una función recursiva que devuelva el factorial de un número pasado 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>	
Ficheros a entregar : ft_iterative_power.c	
Funciones autorizadas : Ninguna	

- Escriba 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.
- 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>	
Ficheros a entregar : ft_recursive_power.c	
Funciones autorizadas : Ninguna	

- Escriba una función recursiva que devuelva una potencia de un número.
- 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_power(int nb, int power);
```

Capítulo VII

Ejercicio 04 : ft_fibonacci

	Ejercicio : 04
	ft_fibonacci
Directorio de entrega : <i>ex04/</i>	
Ficheros a entregar : ft_fibonacci.c	
Funciones autorizadas : Ninguna	


- Escriba 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>	
Ficheros a entregar : ft_sqrt.c	
Funciones autorizadas : Ninguna	

- Escriba 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>	
Ficheros a entregar : ft_is_prime.c	
Funciones autorizadas : Ninguna	

- Escriba 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>	
Ficheros a entregar : ft_find_next_prime.c	
Funciones autorizadas : Ninguna	

- Escriba una función que devuelva el número primo inmediatamente superior o igual al número pasado 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>	
Ficheros a entregar : ft_ten_queens_puzzle.c	
Funciones autorizadas : write	

- Escriba 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.
- 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.