



Piscina C

C 01

Staff 42 [piscine@42.fr](mailto:piscine@42.fr)

*Resumen: Este documento corresponde a la evaluación del módulo C 01 de la piscina de 42.*

# Índice general

I.	Instrucciones	2
II.	Preámbulo	4
III.	Ejercicio 00 : ft_ft	6
IV.	Ejercicio 01 : ft_ultimate_ft	7
V.	Ejercicio 02 : ft_swap	8
VI.	Ejercicio 03 : ft_div_mod	9
VII.	Ejercicio 04 : ft_ultimate_div_mod	10
VIII.	Ejercicio 05 : ft_putstr	11
IX.	Ejercicio 06 : ft_strlen	12
X.	Ejercicio 07 : ft_rev_int_tab	13
XI.	Ejercicio 08 : ft_sort_int_tab	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

El juego del Sirope según Perceval, sacado de la serie *Kaamelott*:

"Bueno, os voy a enseñar las reglas simplificadas, porque las de verdad tienen miga. A ver, la única pega es que, en principio, es un juego de tres. Pero no importa, nos las vamos a apañar.

El objetivo es anunciar valores. Por ejemplo, pongamos que aquí somos tres, se distribuyen tres valores. Digamos: sirope de ocho, sirope de catorce y sirope de veintiuno. De momento no os preocupéis por los siropes. Lo primero que hay que entender son los valores. Si apostáis un valor al principio de una ronda, digamos un sirope de ocho, para empezar por lo bajo, los demás pueden elegir entre pasar o subir la apuesta con un sirope de catorce. Se gira en el sentido de los valores. Por eso, hay que comprender bien el sistema de valores; después todo va sobre ruedas.

Vale, digamos que abro con un sirope de ocho.

Si sois vosotros los que habéis sorbido en la ronda anterior, entonces gira en vuestro sentido. En ese caso, o dejáis pasar diciendo "paso sirope" u os animáis a subir y anunciáis un sirope de catorce. Como hemos iniciado los anuncios, el segundo jugador no tiene derecho a pasar. Podéis o bien apostar un sirope de veintiuno o abandonar vuestro turno diciendo "sirope echado" o "sirope Juanito", depende de la región. Después, o se sigue la partida o ¡hago un "contrasirope"! Y a partir de ahí, sirope de manzana sobre veintiuno y entonces jugamos la partida en cuatro rondas hasta que haya uno que sorba.


Para ganar solo hay tres posibilidades: o anunciáis vuestro sirope de ocho diciendo "buen sirope" y volvemos a contar, o anunciáis vuestro sirope de catorce diciendo "buen sirope, sirope ganador" y se os añade la mitad, o anunciáis vuestro sirope de veintiuno diciendo "buen sirope, medio sirope, sorbido, sirope ganador, sirope cascabel, pasamontañas, sirope rico".

Normalmente se juega con cartas, pero si solo tenéis dados, también se puede jugar con dados porque lo que cuentan son los valores."

Al menos uno de los ejercicios siguientes no tiene ninguna relación con el juego del Sirope.

# Capítulo III

## Ejercicio 00 : ft\_ft


	Ejercicio : 00
	ft_ft
Directorio de entrega : <i>ex00/</i>	
Ficheros a entregar : <b>ft_ft.c</b>	
Funciones autorizadas : Ninguna	

- Escriba una función que reciba como parámetro un puntero a un int y dé al int el valor 42.
- El prototipo de la función deberá ser el siguiente:

```
void      ft_ft(int *nbr);
```

# Capítulo IV

## Ejercicio 01 : ft\_ultimate\_ft

	Ejercicio : 01
	ft_ultimate_ft
	Directorio de entrega : <i>ex01/</i>
	Ficheros a entregar : <b>ft_ultimate_ft.c</b>
	Funciones autorizadas : Ninguna


- Escriba una función que reciba como parámetro un puntero a un puntero a un puntero a un puntero a un puntero a un puntero a un puntero a un puntero a un int y dé al int el valor 42.
- El prototipo de la función deberá ser el siguiente:

```
void      ft_ultimate_ft(int *****nbr);
```



# Capítulo V

## Ejercicio 02 : ft\_swap


	Ejercicio : 02
	ft_swap
	Directorio de entrega : <i>ex02/</i>
	Ficheros a entregar : <b>ft_swap.c</b>
	Funciones autorizadas : Ninguna

- Escriba una función que intercambie el contenido de dos enteros cuyas direcciones se pasan como parámetro.
- El prototipo de la función deberá ser el siguiente:

```
void    ft_swap(int *a, int *b);
```

# Capítulo VI

## Ejercicio 03 : ft\_div\_mod

	Ejercicio : 03
	ft_div_mod
	Directorio de entrega : <i>ex03/</i>
	Ficheros a entregar : <b>ft_div_mod.c</b>
	Funciones autorizadas : Ninguna


- Escriba una función `ft_div_mod` que tenga el prototipo siguiente:

```
void ft_div_mod(int a, int b, int *div, int *mod);
```

- Esta función divide los dos parámetros `a` y `b` y almacena el resultado en el `int` apuntado por `div`.  
También almacena el resto de la división de `a` y `b` en el `int` apuntado por `mod`.

# Capítulo VII

## Ejercicio 04 : ft\_ultimate\_div\_mod

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


- Escriba una función `ft_ultimate_div_mod` que tenga el prototipo siguiente:

```
void ft_ultimate_div_mod(int *a, int *b);
```

- Esta función divide los int apuntados por **a** y **b**.  
El resultado de la división se almacena en el int apuntado por **a**.  
El resultado del resto de la división se almacena en el int apuntado por **b**

# Capítulo VIII

## Ejercicio 05 : ft\_putstr

	Ejercicio : 05
	ft_putstr
	Directorio de entrega : <i>ex05/</i>
	Ficheros a entregar : <b>ft_putstr.c</b>
	Funciones autorizadas : <b>write</b>

- Escriba una función que muestre uno a uno en la pantalla los caracteres de una cadena.
- La dirección del primer carácter de la cadena está incluida en el puntero pasado como parámetro a la función.
- El prototipo de la función deberá ser el siguiente:

```
void    ft_putstr(char *str);
```

# Capítulo IX

## Ejercicio 06 : ft\_strlen


	Ejercicio : 06
	ft_strlen
	Directorio de entrega : <i>ex06/</i>
	Ficheros a entregar : <b>ft_strlen.c</b>
	Funciones autorizadas : Ninguna

- Escriba una función que cuente el número de caracteres de una cadena de caracteres y que devuelva el número encontrado.
- El prototipo de la función deberá ser el siguiente:

```
int  ft_strlen(char *str);
```

# Capítulo X

## Ejercicio 07 : ft\_rev\_int\_tab


	Ejercicio : 07
	ft_rev_int_tab
	Directorio de entrega : <i>ex07/</i>
	Ficheros a entregar : <b>ft_rev_int_tab.c</b>
	Funciones autorizadas : Ninguna

- Escriba una función que invierta el orden de los elementos de una tabla de enteros.
- Los parámetros son un puntero a entero y el número de enteros de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
void    ft_rev_int_tab(int *tab, int size);
```

# Capítulo XI

## Ejercicio 08 : ft\_sort\_int\_tab

	Ejercicio : 08
	ft_sort_int_tab
	Directorio de entrega : <i>ex08/</i>
	Ficheros a entregar : <b>ft_sort_int_tab.c</b>
	Funciones autorizadas : Ninguna

- Escriba una función que ordene una tabla de enteros en orden creciente.
- Los parámetros son un puntero a entero y el número de enteros de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
void    ft_sort_int_tab(int *tab, int size);
```