



C Piscine

C 01

Resumen: Este documento corresponde al enunciado del módulo C 01 de la C Piscine de 42.

Versión: 5.3

Índice general

I.	Instrucciones	2
II.	Introducción	4
III.	Ejercicio 00 : ft_ft	5
IV.	Ejercicio 01 : ft_ultimate_ft	6
V.	Ejercicio 02 : ft_swap	7
VI.	Ejercicio 03 : ft_div_mod	8
VII.	Ejercicio 04 : ft_ultimate_div_mod	9
VIII.	Ejercicio 05 : ft_putstr	10
IX.	Ejercicio 06 : ft_strlen	11
X.	Ejercicio 07 : ft_rev_int_tab	12
XI.	Ejercicio 08 : ft_sort_int_tab	13
XII.	Entrega y evaluación	14

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.



No te olvides de añadir el *header estándar de 42* en cada uno de tus archivos `.c/.h`. ¡La norminette comprueba siempre que esté presente!



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

Capítulo II

Introducción


Diálogo de la película *Una Noche en la Ópera*, de los Heranos Marx:

- Haga el favor de poner atención en la primera cláusula porque es muy importante. Dice que... la parte contratante de la primera parte será considerada como la parte contratante de la primera parte. ¿Qué tal, está muy bien, eh?
- No, eso no está bien. Quisiera volver a oírlo.
- Dice que... la parte contratante de la primera parte será considerada como la parte contratante de la primera parte.
- Esta vez creo que suena mejor.
- Si quiere se lo leo otra vez.
- Tan solo la primera parte.
- ¿Sobre la parte contratante de la primera parte?
- No, solo la parte de la parte contratante de la primera parte.
- Oiga, ¿por qué hemos de pelearnos por una tontería como ésta? La cortamos.
- Sí, es demasiado largo. ¿Qué es lo que nos queda ahora?
- Dice ahora... la parte contratante de la segunda parte será considerada como la parte contratante de la segunda parte.
- Eso si que no me gusta nada. Nunca segundas partes fueron buenas. Escuche: ¿por qué no hacemos que la primera parte de la segunda parte contratante sea la segunda parte de la primera parte?.

Al menos uno de los ejercicios siguientes no tiene ninguna relación con este diálogo

Capítulo III

Ejercicio 00 : ft_ft


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

- Escribe 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>
	Archivos a entregar: ft_ultimate_ft.c
	Funciones autorizadas: Ninguna

- Escribe 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 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>	
Archivos a entregar: ft_swap.c	
Funciones autorizadas: Ninguna	

- Escribe una función que intercambie el contenido de dos enteros cuyas direcciones se utilicen 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>	
Archivos a entregar: ft_div_mod.c	
Funciones autorizadas: Ninguna	


- Escribe 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>	
Archivos a entregar: ft_ultimate_div_mod.c	
Funciones autorizadas: Ninguna	


- Escribe 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>	
Archivos a entregar: ft_putstr.c	
Funciones autorizadas: write	

- Escribe una función que muestre uno a uno en la pantalla los caracteres de un string.
- La dirección del primer carácter del string está incluida en el puntero utilizado como parámetro para 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>	
Archivos a entregar: ft_strlen.c	
Funciones autorizadas: Ninguna	

- Escribe una función que cuente el número de caracteres de un string 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>
	Archivos a entregar: ft_rev_int_tab.c
	Funciones autorizadas: Ninguna

- Escribe una función que invierta un array de int (el primer elemento va el último, etc).
- Los parámetros son un puntero a int y el número de int en el array.
- 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>
	Archivos a entregar: ft_sort_int_tab.c
	Funciones autorizadas: Ninguna

- Escribe una función que ordene un array de int en orden ascendente.
- Los parámetros son un puntero a int y el número de int en el array.
- El prototipo de la función deberá ser el siguiente:

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

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.