



C Piscine

C 11

*Resumen: Este documento corresponde al enunciado del módulo C 11 de la C Piscine de 42.*

*Versión: 6.2*

# Índice general

I.	Instrucciones	2
II.	Introducción	4
III.	Ejercicio 00 : ft_foreach	5
IV.	Ejercicio 01 : ft_map	6
V.	Ejercicio 02 : ft_any	7
VI.	Ejercicio 03 : ft_count_if	8
VII.	Ejercicio 04 : ft_is_sort	9
VIII.	Ejercicio 05 : do-op	10
IX.	Ejercicio 06 : ft_sort_string_tab	12
X.	Ejercicio 07 : ft_advanced_sort_string_tab	13
XI.	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, sé 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.

# Capítulo II

## Introducción

Cita sacada de la película V de Vendetta:


¡Vedà! A primera Vista, un humilde Veterano de VaudeVille,  
en el papel de Víctima y Villano por Vicisitudes de la Vida.  
Este que Ves, ya no mas Velo de Vanidad, es un Vestigio de  
la Vox populi, ahora Vacua, desVanecida. Sin embargo, esta Valerosa  
Visión de una extinta Vejación se siente reViVida y ha hecho Voto de  
Vencer el Vil Veneno de estas Víboras en aVanzada que Velan por los  
Violentos Viciosos y por la Violación de la Voluntad. El único Veredicto  
es Venganza; Vendetta, como Voto, y no en Vano, pues la Valía y Veracidad  
de esta un día Vindicará al Vigilante y al Virtuoso. La verdad, esta  
Vichyssoise de Verborrea se está VolViendo muy Verbosa así que solo añadiré  
que es un Verdadero placer conocerte y que puedes llamarme V.



Evite las aliteraciones. Siempre.

# Capítulo III

## Ejercicio 00 : ft\_foreach

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

- Escribe una función `ft_foreach` que, para una tabla de enteros determinada, aplique una función sobre cada uno de los elementos de la tabla. Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:


```
void      ft_foreach(int *tab, int length, void(*f)(int));
```

- Por ejemplo, para mostrar todos los enteros de la tabla, se podrá llamar a la función `ft_foreach` de la siguiente forma:

```
ft_foreach(tab, 1337, &ft_putnbr);
```

# Capítulo IV

## Ejercicio 01 : ft\_map


	Ejercicio: 01
ft_map	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: <b>ft_map.c</b>	
Funciones autorizadas: <b>malloc</b>	

- Escribe una función **ft\_map** que, para una tabla de enteros determinada, aplique una función sobre cada uno de los elementos de la tabla (por orden) y retorne una tabla con todos los valores de retorno.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
int      *ft_map(int *tab, int length, int(*f)(int));
```

# Capítulo V

## Ejercicio 02 : ft\_any

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

- Escribe una función **ft\_any** que devuelva 1 si, al menos uno de los elementos de la tabla pasados a la función **f** devuelve algo diferente de 0, si no devolverá 0.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:


```
int ft_any(char **tab, int(*f)(char*));
```

- La tabla terminará con un puntero NULL.



# Capítulo VI

## Ejercicio 03 : ft\_count\_if


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

- Escribe una función `ft_count_if` que devuelva el número de elementos de la tabla que, cuando sean pasados a la función `f`, no devuelvan 0.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

# Capítulo VII

## Ejercicio 04 : ft\_is\_sort


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

- Escribe una función `ft_is_sort` que devuelva 1 si la tabla está ordenada y 0 en el caso contrario.
- La función pasada como parámetro devolverá un entero negativo si el primer argumento es inferior al segundo, 0 si son iguales y un entero positivo en cualquier otro caso.
- El prototipo de la función deberá ser el siguiente:

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

# Capítulo VIII

## Ejercicio 05 : do-op

	Ejercicio: 05
do-op	
Directorio de entrega: <i>ex05/</i>	
Archivos a entregar: Todos los archivos necesarios para su programa	
Funciones autorizadas: <b>write</b>	

- Crea un programa que se llame do-op.
- El programa deberá ser ejecutado con tres argumentos: do-op valor1 operador valor2
- Ejemplo:

```
$> ./do-op 42 "+" 21
63
$>
```

- Deberías usar una tabla de punteros a funciones para llamar a la función que corresponda a un operador.
- En el caso de un operador desconocido, tu programa debe mostrar 0.
- Si el número de argumentos no es correcto, do-op no muestra nada.
- Tu programa debe aceptar y mostrar el resultado con los siguientes operadores: '+', '-', '/', '\*', y '%'
- Tu programa debe tratar el valor el valor como int.
- En el caso de una división por 0 tu programa debe mostrar:

```
Stop : division by zero
```

- En el caso de un módulo por 0 tu programa debe mostrar:


```
Stop : modulo by zero
```

- He aquí un ejemplo de pruebas de la Moulinette:

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

# Capítulo IX

## Ejercicio 06 : ft\_sort\_string\_tab


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

- Escribe la función `ft_sort_string_tab` que ordene por código ASCII las cadenas de caracteres.
- `tab` se terminará con un `NULL`.
- La clasificación se realizará intercambiando los punteros de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
void ft_sort_string_tab(char **tab);
```

# Capítulo X

## Ejercicio 07 : ft\_advanced\_sort\_string\_tab

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

- Escribe la función `ft_advanced_sort_string_tab` que ordene, en función del retorno de la función pasada como parámetro.
- La clasificación se realizará intercambiando los punteros de la tabla.
- `tab` se terminará con un `NULL`.
- El prototipo de la función deberá ser el siguiente:

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



La llamada a `ft_advanced_sort_string_tab()` con un segundo parámetro `ft_strcmp` dará el mismo resultado que `ft_sort_string_tab()`.

# Capítulo XI

## 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.