



Piscina C

C 08

Staff 42 pedago@42.fr

Resumen: Este documento corresponde a la evaluación del módulo C 08 de la piscina de 42.

Índice general

I.	Instrucciones	2
II.	Preámbulo	4
III.	Ejercicio 00 : ft.h	5
IV.	Ejercicio 01 : ft_boolean.h	6
V.	Ejercicio 02 : ft_abs.h	8
VI.	Ejercicio 03 : ft_point.h	9
VII.	Ejercicio 04 : ft_strs_to_tab	10
VIII.	Ejercicio 05 : ft_show_tab	12

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

Esto es lo que nos cuenta la enciclopedia colaborativa *Wikipedia* sobre el ornitorrinco:

El ornitorrinco (*Ornithorhynchus anatinus*) es una especie de pequeño mamífero semiacuático endémico del este de Australia, incluida Tasmania. Es una de las cinco especies del orden de los monotremas, único orden de mamíferos que ponen huevos en lugar de parir crías completamente formadas (las otras cuatro especies son equidnas). Es la única especie superviviente de la familia Ornithorhynchidae y del género Ornithorhynchus, aunque se hayan descubierto una gran cantidad de fragmentos de especies fósiles de esta familia y de este género.

La apariencia extraña de este mamífero ponedor de huevos, dotado de espolones venenosos, con la mandíbula córnea parecida al pico de un pato, una cola que recuerda a un castor, que le sirve tanto de timón en el agua como de reserva de grasa, y patas de nutria sorprendió enormemente a los primeros exploradores que lo descubrieron; buena parte de los naturalistas europeos pensaron que se trataba de una broma. Es uno de pocos mamíferos venenosos: el macho lleva sobre las patas traseras un espolón que puede liberar veneno capaz de provocar un dolor intenso a un humano. Las características singulares del ornitorrinco lo convierten en un sujeto de estudio importante para entender mejor la evolución de las especies animales y se ha convertido en uno de los símbolos de Australia:


ha sido utilizado como mascota en acontecimientos nacionales y aparece en el dorso de la moneda australiana de 20 céntimos.

Fue cazado hasta principios del siglo XX por su piel, pero actualmente está protegido. A pesar de que los programas de reproducción en cautividad han tenido un éxito bastante limitado y de que sea un animal vulnerable a los efectos de la contaminación, todavía no se considera que la especie esté en peligro.

Este tema no tiene nada que ver con el ornitorrinco.

Capítulo III

Ejercicio 00 : ft.h


	Ejercicio : 00
	ft.h
	Directorio de entrega : <i>ex00/</i>
	Ficheros a entregar : ft.h
	Funciones autorizadas : Ninguna

- Escriba su archivo `ft.h`
- Contiene todos los prototipos de las funciones:

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strncmp(char *s1, char *s2);
```

Capítulo IV

Ejercicio 01 : ft_boolean.h

	Ejercicio : 01
	ft_boolean.h
	Directorio de entrega : <i>ex01/</i>
	Ficheros a entregar : ft_boolean.h
	Funciones autorizadas : Ninguna

- Escriba un archivo `ft_boolean.h` que haga que compile y funcione correctamente el main siguiente:

```
#include "ft_boolean.h"

void      ft_putstr(char *str)
{
    while (*str)
        write(1, str++, 1);
}

t_bool     ft_is_even(int nbr)
{
    return ((EVEN(nbr)) ? TRUE : FALSE);
}

int        main(int argc, char **argv)
{
    (void)argv;
    if (ft_is_even(argc - 1) == TRUE)
        ft_putstr(EVEN_MSG);
    else
        ft_putstr(ODD_MSG);
    return (SUCCESS);
}
```

- Este programa mostrará:

```
I have an even number of arguments.
```


- O

```
I have an odd number of arguments.
```

- seguido de un salto de línea, en el caso apropiado.

Capítulo V

Ejercicio 02 : ft_abs.h


	Ejercicio : 02
	ft_abs.h
	Directorio de entrega : <i>ex02/</i>
	Ficheros a entregar : ft_abs.h
	Funciones autorizadas : Ninguna

- Escriba una macro ABS que sustituya su parámetro por su valor absoluto:

```
#define ABS(Value)
```

Capítulo VI

Ejercicio 03 : ft_point.h

	Ejercicio : 03
	ft_point.h
	Directorio de entrega : <i>ex03/</i>
	Ficheros a entregar : ft_point.h
	Funciones autorizadas : Ninguna

- Escriba un archivo `ft_point.h` que haga que compile el main siguiente:

```
#include "ft_point.h"


void      set_point(t_point *point)
{
    point->x = 42;
    point->y = 21;
}

int       main(void)
{
    t_point      point;

    set_point(&point);
    return (0);
}
```

Capítulo VII

Ejercicio 04 : ft_strs_to_tab

	Ejercicio : 04
	ft_strs_to_tab
	Directorio de entrega : ex04/
	Ficheros a entregar : ft_strs_to_tab.c
	Funciones autorizadas : malloc, free

- Escriba una función que reciba como parámetro una tabla de cadenas de caracteres al igual que el tamaño de la tabla y devuelva una tabla de estructuras.
- El prototipo de la función deberá ser el siguiente:

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- La función debe transformar cada elemento de la tabla de cadenas de caracteres en estructura.
- La estructura será definida en el archivo ft_stock_str.h de la siguiente forma:


```
typedef struct s_stock_str
{
    int size;
    char *str;
    char *copy;
} t_stock_str;
```

- size es el tamaño de la cadena de caracteres;
 - str es la cadena de caracteres;
 - copy es una copia de la cadena de caracteres;
- Debe conservar el orden de los elementos de av.

- Se tendrá que asignar memoria a la tabla de estructuras y el último elemento tendrá 0 como valor de `str`, para indicar el final de la tabla.
- Si surge un error de asignación de memoria la función deberá retornar un puntero NULL.
- Probaremos su función con nuestro `ft_show_tab` (ejercicio siguiente). ¡Haga lo que sea preciso para que funcione!

Capítulo VIII

Ejercicio 05 : ft_show_tab

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

- Escriba una función que muestre el contenido de una tabla creada con la función anterior.
- El prototipo de la función deberá ser el siguiente:

```
void ft_show_tab(struct s_stock_str *par);
```

- La estructura es la misma que la del ejercicio anterior y estará en el archivo `ft_stock_str.h` que le proporcionaremos.
- Para cada elemento de la tabla:
 - la cadena de caracteres irá seguida de un salto de línea
 - el tamaño irá seguido de un salto de línea
 - la copia de la cadena de caracteres (que podrá haber sido modificada) irá seguida de un salto de línea
- Probamos su función con nuestro `ft_strs_to_tab` (ejercicio anterior). ¡Haga lo que sea preciso para que funcione!