



Piscina C

C 09

Resumen: Este documento corresponde a la evaluación del módulo C 09 de la piscina de 42.

Índice general

| | | |
|------|-------------------------|---|
| I. | Instrucciones | 2 |
| II. | Preámbulo | 4 |
| III. | Ejercicio 00 : libft | 5 |
| IV. | Ejercicio 01 : Makefile | 6 |
| V. | Ejercicio 02 : ft_split | 8 |

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

Diálogo sacado de la película El gran Lebowski:

Walter: Lo siento Smokey, has pisado la raya: es falta.
Smokey: ¡Una mierda! Apúntame ocho, Nota...
Walter: Disculpa, apunta un cero, siguiente tú.
Smokey: Y una mierda Walter, apúntame un ocho Nota...
Walter: Smokey esto no es Vietnam, en los bolos hay reglas...
El Nota: Eh, Walter, venga hombre, es Smokey tío: pisó un poquito con el dedo gordo, vale... ¡No es más que un juego, tío!
Walter: Estamos jugando una liga, aquí se decide quién pasa a la siguiente ronda, ¿me equivoco?
Smokey: Sí, pero tampoco he...
Walter: ¿Me equivoco?
Smokey: Ya, pero no he pisado... Pásame el marcador, Nota... me voy a apuntar un ocho.
Walter: [Saca su pistola] Smokey, amigo, estás a punto de conocer el dolor...
El Nota: ¡No te pases, tío!
Walter: [Amenazante] ¡Como te apuntes un ocho, vas a conocer el dolor!
Smokey: Tío, yo no...
Walter: ¡Vas a conocer el dolor!
Smokey: ¿Nota? Este tío es tu compañero...
Walter: ¿Es que todo el mundo se ha VUELTO LOCO? [Se levanta]
¿ES QUE SOY EL ÚNICO POR AQUÍ QUE NO SE CAGA EN LAS REGLAS?
¡APUNTA UN CERO!
El Nota: Tío, están llamando a la pasma, guarda la pipa...
Walter: [Apunta su pistola sobre Smokey] ¡APUNTA UN CERO!
El Nota: Walter, guarda la pipa...
Smokey: ...
El Nota: Walter...
Walter: [Arma su pistola] ¿CREÉIS QUE ESTOY DE COÑA?
¡APUNTA UN CERO!
Smokey: De acuerdo, me he puesto un cero... ¿Ya estás contento? ¡Capullo!
Walter: [Se tranquiliza]... Una liga es una liga.

Capítulo III

Ejercicio 00 : libft

| | |
|---|----------------|
|  | Ejercicio : 00 |
| libft | |
| Directorio de entrega : <i>ex00/</i> | |
| Ficheros a entregar : <code>libft_creator.sh</code> , <code>ft_putchar.c</code> , <code>ft_swap.c</code> , <code>ft_putstr.c</code> , <code>ft_strlen.c</code> , <code>ft_strcmp.c</code> | |
| Funciones autorizadas : <code>write</code> | |

- Cree su biblioteca `ft`. Se llamará `libft.a`.
- Un script shell denominado `libft_creator.sh` compilará como es debido los archivos fuente y creará su biblioteca.
- Esta biblioteca debe incluir todas las funciones que se indican a continuación:


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

- Ejecutaremos el comando siguiente:

```
sh libft_creator.sh
```

Capítulo IV

Ejercicio 01 : Makefile

| | |
|---|----------------|
|  | Ejercicio : 01 |
| Makefile | |
| Directorio de entrega : <i>ex01/</i> | |
| Ficheros a entregar : Makefile | |
| Funciones autorizadas : Ninguna | |

- Escriba el **Makefile** que compile una librería **libft.a**.
- Su makefile debe mostrar cada comando que ejecute sin ningún tipo de floritura.
- Su makefile no debe ejecutar ningún comando inútil.
- El **Makefile** irá a buscar los archivos fuente al directorio **srcs**.
- Estos archivos fuente serán: **ft_putchar.c**, **ft_swap.c**, **ft_putstr.c**, **ft_strlen.c**, **ft_strcmp.c**
- El **Makefile** irá a buscar los archivos headers al directorio **includes**.
- Estos archivos headers serán: **ft.h**
- Tendrá que compilar los archivos **c** utilizando **gcc** y las opciones flag **-Wall -Wextra -Werror**, en ese orden.
- La librería estará en la raíz del ejercicio.
- Los archivos **.o** tendrán que estar al lado de sus archivos **.c** correspondientes.
- El **Makefile** también tendrá que implementar reglas **clean**, **fclean**, **re**, la regla **all** y, por supuesto, **libft.a**.
- Ejecutar solo **make** debe ser equivalente a **make all**.
- La regla **all** tendrá que hacer lo mismo que **make libft.a**.
- La regla **clean** tendrá que retirar todos los archivos temporales que se hayan generado.


- La regla `fclean` hará lo mismo que un `make clean` y borrará también el binario creado al ejecutar `make all`.
- La regla `re` hará lo mismo que un `make fclean` seguido de un `make all`.
- Su makefile no debe recompilar los archivos inútilmente.
- Solo recogeremos su Makefile y lo probaremos con nuestros archivos.



¡Tenga cuidado con los wildcards!

Capítulo V

Ejercicio 02 : ft_split

| | |
|---|----------------|
|  | Ejercicio : 02 |
| | ft_split |
| Directorio de entrega : <i>ex02/</i> | |
| Ficheros a entregar : ft_split.c | |
| Funciones autorizadas : malloc | |

- Escriba una función que divida una cadena de caracteres en función de otra cadena de caracteres.
- Habrá que utilizar cada carácter de la cadena **charset** como separador.
- La función devuelve una tabla donde cada uno de sus elementos contiene la dirección de una cadena de caracteres comprendida entre dos separadores. El último elemento de la tabla tendrá que ser igual a 0 para indicar el final de la tabla.
- Su tabla no puede tener cadenas vacías. Saque las conclusiones pertinentes.
- La cadena que se transmitirá no será modificable.
- El prototipo de la función deberá ser el siguiente:

```
char **ft_split(char *str, char *charset);
```