

C Piscine C 07

Resumen: Este documento corresponde al enunciado del módulo C 07 de la C Piscine de 42.

Versión: 3

# Índice general

1.	Instrucciones	2
II.	Introducción	4
III.	Ejercicio 00 : ft_strdup	5
IV.	Ejercicio 01 : ft_range	6
V.	Ejercicio 02 : ft_ultimate_range	/7
VI.	Ejercicio 03 : ft_strjoin	8
VII.	Ejercicio 04 : ft_convert_base	9
VIII.	Ejercicio 05 : ft split	10

#### 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 gcc.
- Si tu programa no compila, tendrán un 0.
- <u>No puedes</u> dejar en tu directorio <u>ningún</u> archivo que no se haya indicado de forma <u>explícita</u> en los enunciados de los <u>ejercicios</u>.
- ¿Tienes alguna pregunta? Pregunta a tu compañero de la derecha. Si no, prueba con tu compañero de la izquierda.

- $\bullet$  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.



Para este módulo, la Norminette debe ser ejecutada con el flag -R CheckForbiddenSourceHeader. La Moulinette también lo utilizará.

# Capítulo II

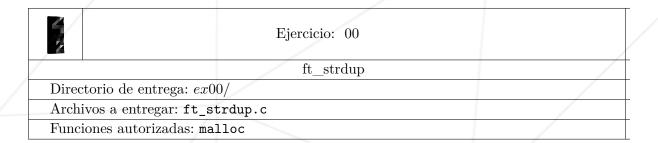
#### Introducción

 $\mbox{He}$  aquí una lista de los monstruos que se pueden encontrar en las famosas mazmorras de Naheulbeuk:

- Todo tipo de zombis;
- Arañas gigantes;
- Orcos;
- Goblins;
- Trolls de las cavernas;
- Brujos;
- Guerreros malditos;
- Ratas mutantes;
- Una botella de aceite;
- Papel higiénico;
- Dos esponjas;
- Raviolis.

# Capítulo III

Ejercicio 00 : ft\_strdup

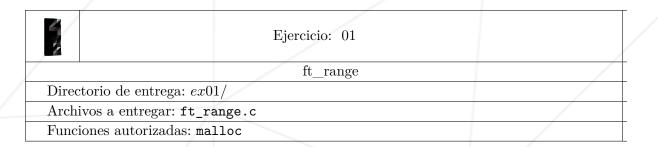


- Reproduce el comportamiento de la función strdup (man strdup).
- El prototipo de la función deberá ser el siguiente:

char \*ft\_strdup(char \*src);

### Capítulo IV

## Ejercicio 01 : ft\_range



- Escribe una función ft\_range que devuelva un array de int. Este array de int contendrá todos los valores entre min y max.
- Min incluido max excluido.
- El prototipo de la función deberá ser el siguiente:

```
int *ft_range(int min, int max);
```

• Si el valor min es superior o igual al valor max, se devolverá un puntero nulo.

#### Capítulo V

### Ejercicio 02: ft\_ultimate\_range

/		
	Ejercicio: 02	
	ft_ultimate_range	
Directo	rio de entrega: $ex02/$	/
Archivos a entregar: ft_ultimate_range.c		/
Funciones autorizadas: malloc		/

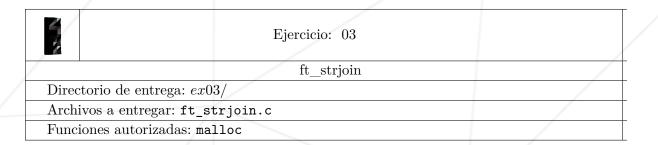
- Escribe una función ft\_ultimate\_range que asigne memoria y datos a un array de int. Este array de int contendrá todos los valores entre min y max.
- Min incluido max excluido.
- El prototipo de la función deberá ser el siguiente:

```
int ft_ultimate_range(int **range, int min, int max);
```

- Devolverá el tamaño de range (o -1 en caso de problemas).
- Si el valor min es superior o igual al valor max, range apuntará a NULL y retornará 0.

### Capítulo VI

### Ejercicio 03: ft\_strjoin

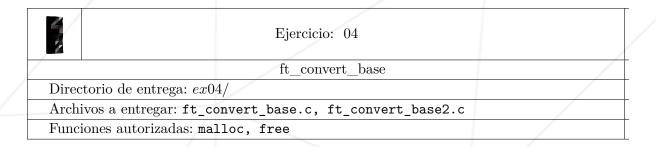


- Escribe una función que concatene el conjunto de cadenas de caracteres apuntadas por strs, separándolas por sep.
- size representa el tamaño de strs.
- $\bullet\,$  Si size vale 0, habrá que retornar una cadena de caracteres vacía liberable.
- El prototipo de la función deberá ser el siguiente:

char \*ft\_strjoin(int size, char \*\*strs, char \*sep);

#### Capítulo VII

#### Ejercicio 04 : ft\_convert\_base

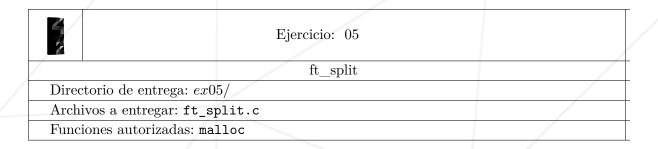


- Escribe una función que devuelva el resultado de la conversión de la cadena nbr, expresada en una base base\_from, a una base base\_to.
- nbr, base\_from, base\_to no son necesariamente modificables.
- $\bullet$  nbr seguirá las mismas reglas que ft\_atoi\_base. Por lo tanto, cuidado con el + , y con los espacios blancos.
- El número representado por nbr cabe en un int.
- Si una base es incorrecta, la función retornará NULL.
- $\bullet$  Si es preciso que el número devuelto lleve un prefijo, este será un solo y único , sin espacios ni + .
- El prototipo de la función deberá ser el siguiente:

char \*ft\_convert\_base(char \*nbr, char \*base\_from, char \*base\_to);

#### Capítulo VIII

Ejercicio 05: ft\_split



- Escribe 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 retorna un array donde cada uno de sus elementos contiene la dirección de una cadena de caracteres comprendida entre dos separadores. El último elemento del array tendrá que ser igual a 0 para indicar el final del array.
- Tu array no puede tener cadenas vacías. Saca las conclusiones pertinentes.
- La cadena pasada como argumento no será modificable.
- El prototipo de la función deberá ser el siguiente:

char \*\*ft\_split(char \*str, char \*charset);