

Piscina C C 07

Staff 42 pedago@42.fr

Resumen: Este documento corresponde a la evaluación del módulo C 07 de la piscina de 42.

Índice general

1.	Instrucciones	2
II.	Preámbulo	4
III.	Ejercicio 00 : ft_strdup	5
IV.	Ejercicio 01 : ft_range	6
V.	Ejercicio 02 : ft_ultimate_range	77
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 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.
- <u>No debe</u> dejar en su directorio <u>ningún</u> 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

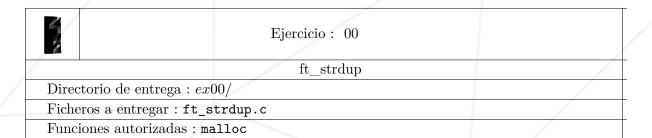
Preámbulo

 \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

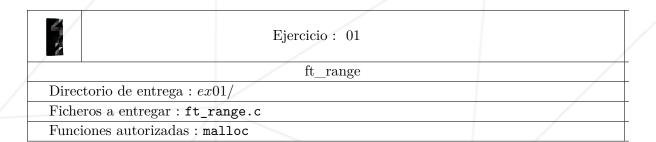


- Reproducir de manera idéntica el funcionamiento de la función strdup (man strdup).
- El prototipo de la función deberá ser el siguiente:

char *ft_strdup(char *src);

Capítulo IV

Ejercicio 01 : ft_range



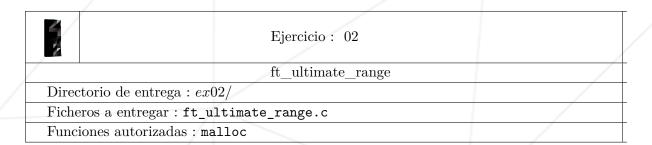
- Escriba una función ft_range que devuelva un array de int. Este array de int contendrá todos los valores entre min y max.
- Min incluido max excluido.
- El prototipo de la función deberá ser el siguiente:

```
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



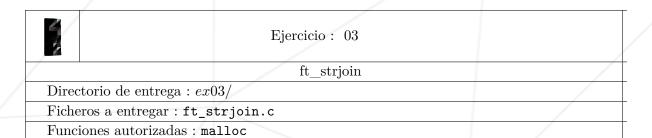
- Escriba una función ft_ultimate_range que asigne memoria y datos a un array de int. Este array de int contendrá todos los valores entre min y max.
- Min incluido max excluido.
- El prototipo de la función deberá ser el siguiente:

```
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

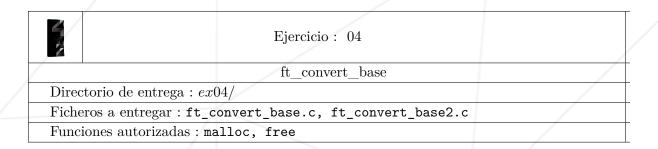


- Escriba una función que concatene el conjunto de cadenas de caracteres apuntadas por strs, separándolas por sep.
- size representa el tamaño de strs.
- Si size vale 0, habrá que retornar una cadena de caracteres vacía que podemos free().
- El prototipo de la función deberá ser el siguiente:

char *ft_strjoin(int size, char **strs, char *sep);

Capítulo VII

Ejercicio 04 : ft_convert_base

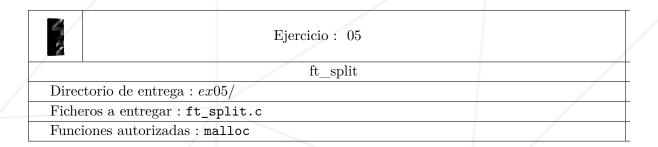


- Escriba una función que devuelva el resultado de la conversión de la cadena nbr, expresada en una base base_from, a una base base_to.
- nbr, base_from, base_to no son necesariamente modificables.
- nbr seguirá las mismas reglas que ft_atoi_base. Por lo tanto, cuidado con el '+', '-' y con los espacios blancos.
- El número representado por nbr cabe en un int.
- Si una base es incorrecta, la función retornará NULL.
- 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



- 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 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.
- Su array 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);