



Piscina C

C 04

Staff 42 pedago@42.fr

Resumen: Este documento corresponde a la evaluación del módulo C 04 de la piscina de 42.

Índice general

I.	Instrucciones	2
II.	Preámbulo	4
III.	Ejercicio 00 : ft_strlen	5
IV.	Ejercicio 01 : ft_putstr	6
V.	Ejercicio 02 : ft_putnbr	7
VI.	Ejercicio 03 : ft_atoi	8
VII.	Ejercicio 04 : ft_putnbr_base	9
VIII.	Ejercicio 05 : ft_atoi_base	11

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

He aquí la letra de los créditos de Nicky Larson:

En la noche una sombra huye
Es un asesino que se escapa
Como un demonio sonríe
Su crimen quedará impune
Aparece un coche
Un frenazo, las ruedas crujen
Un disparo que resuena
La justicia se llama Nicky

[Estribillo]
En el calor
De la noche
El mal siempre será castigado
Ningún peligro le impresiona
Le atraen los golpes duros
Y la justicia le apasiona
Nicky Larson no le teme a nadie
Cuando los disparos retumban
Se mueve como un rayo
Sobre todo si la chica es guapa
Nicky Larson no le teme a nadie

Como un cazador persigue a su presa
Para que la justicia y el derecho
Triunfen, está dispuesto a dar
Su vida sin dudarlo
Cuando aparece su silueta
Los malos se ponen a temblar
Saben que nunca podrán
Escapar a este justiciero

[Estribillo]

Desgraciadamente, estos enunciados no tienen nada que ver con Nicky Larson.

Capítulo III

Ejercicio 00 : ft_strlen


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

- Escriba una función que cuente el número de caracteres de una cadena de caracteres y que devuelva el número encontrado.
- El prototipo de la función deberá ser el siguiente:

```
int  ft_strlen(char *str);
```

Capítulo IV

Ejercicio 01 : ft_putstr


	Ejercicio : 01
	ft_putstr
	Directorio de entrega : <code>ex01/</code>
	Ficheros a entregar : <code>ft_putstr.c</code>
	Funciones autorizadas : <code>write</code>

- Escriba una función que muestre uno a uno en la pantalla los caracteres de una cadena.
- La dirección del primer carácter de la cadena está incluida en el puntero pasado como parámetro a la función.
- El prototipo de la función deberá ser el siguiente:

```
void    ft_putstr(char *str);
```

Capítulo V

Ejercicio 02 : ft_putnbr

	Ejercicio : 02
	ft_putnbr
	Directorio de entrega : <i>ex02/</i>
	Ficheros a entregar : ft_putnbr.c
	Funciones autorizadas : write


- Escriba una función que muestre un número pasado como parámetro. La función deberá ser capaz de mostrar todos los valores posibles en una variable de tipo `int`.
- El prototipo de la función deberá ser el siguiente:

```
void ft_putnbr(int nb);
```

- Por ejemplo:
 - `ft_putnbr(42)` muestra "42".

Capítulo VI

Ejercicio 03 : ft_atoi

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

- Escriba una función que convierta el principio de la cadena apuntada por str en un entero de tipo int
- str puede empezar con un número arbitrario de espacios (tal y como lo define `isspace(3)`)
- A continuación, str puede ir seguida de un número arbitrario de signos + y de signos -. El signo - hará cambiar el signo del entero devuelto en función del número de signos - y si este es par o impar.
- Para terminar, str tendrá que estar compuesta por dígitos en base 10
- Su función tendrá que leer str, siempre que esta última cumpla con las reglas mencionadas anteriormente, y tendrá que devolver el número encontrado hasta entonces.
- No debería tener en cuenta los desbordamientos (overflows y underflows), en estos casos el resultado se considera indefinido.
- Puede comparar su función con la verdadera función atoi, quitando la parte de los signos y del overflow.
- He aquí el ejemplo de un programa que muestra el valor devuelto por atoi:


```
$>./a.out " ----++1234ab567"  
-1234
```

- El prototipo de la función deberá ser el siguiente:

```
int    ft_atoi(char *str);
```

Capítulo VII

Ejercicio 04 : ft_putnbr_base


	Ejercicio : 04
	ft_putnbr_base
	Directorio de entrega : <i>ex04/</i>
	Ficheros a entregar : ft_putnbr_base.c
	Funciones autorizadas : write

- Escriba una función que muestre un número en el terminal en una base determinada.
- Se proporciona este número en forma de un `int` y la base en forma de una **cadena de caracteres**.
- La base contiene todos los símbolos que se pueden utilizar par mostrar el número:
 - 0123456789 es la base que se suele utilizar para representar nuestros números decimales;
 - 01 es una base binaria;
 - 0123456789ABCDEF es una base hexadecimal;
 - poniguay es una base octal.
- La función debe gestionar números negativos.
- Si un parámetro contiene un error la función no muestra nada. Un error puede ser:
 - la base está vacía o su tamaño es 1;
 - la base contiene dos veces el mismo carácter;
 - la base contiene los caracteres + o -.
- El prototipo de la función deberá ser el siguiente:

```
void      ft_putnbr_base(int nbr, char *base);
```

Capítulo VIII

Ejercicio 05 : ft_atoi_base

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

- Escriba una función que convierta el principio de la cadena apuntada por str en un entero de tipo int.
- str está en una base específica pasada como segundo argumento de la función.
- Con excepción del sistema de base, esta función debe reproducir el comportamiento de ft_atoi.
- Si un parámetro contiene un error, la función devuelve 0. Un error puede ser:
 - la base está vacía o su tamaño es 1;
 - la base contiene dos veces el mismo carácter;
 - la base contiene los caracteres + o - o espacios;
- El prototipo de la función deberá ser el siguiente:

```
int      ft_atoi_base(char *str, char *base);
```