



C Piscine

C 04

Resumen: Este documento corresponde al enunciado del módulo C 04 de la C Piscine de 42.

Versión: 4.3

Índice general

I.	Instrucciones	2
II.	Introducción	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	10
IX.	Entrega y evaluación	11

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

- 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

He aquí la letra de los créditos de Teenage Mutant Ninja Turtles:

Teenage Mutant Ninja Turtles
Teenage Mutant Ninja Turtles
Teenage Mutant Ninja Turtles
Heroes in a Half-shell Turtle Power

Here we go
It's the lean, green, ninja team
On the scene, cool teens doin' ninja things
So extreme, out the sewer, like laser beams
Get rocked with the Shell-shocked Pizza Kings

Can't stop these radical dudes
The secret of the ooze made the chosen few
Emerge from the shadows, to make their moves
The good guys win and the bad guys lose

Leonardo's the leader in blue
Does anything it takes to get his ninjas through
Donatello is a fellow, has a way with machines
Raphael has the most attitude on the team


Michelangelo, he's one of a kind
And you know just where to find him when it's party time
Master Splinter taught them every single skill they need
To be one lean, mean, green, incredible team

Teenage Mutant Ninja Turtles
Teenage Mutant Ninja Turtles
Teenage Mutant Ninja Turtles
Heroes in a Half-shell Turtle Power

Desgraciadamente, el enunciado de este proyecto no tiene nada que ver con Teenage Mutant Ninja Turtles.

Capítulo III

Ejercicio 00 : ft_strlen


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

- Escribe una función que cuente el número de caracteres de un string 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: <i>ex01/</i>
	Archivos a entregar: ft_putstr.c
	Funciones autorizadas: write

- Escribe una función que muestre uno a uno en la pantalla los caracteres de una cadena de caracteres.
- La dirección del primer carácter de la cadena de caracteres está incluida en el puntero usado 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>	
Archivos a entregar: ft_putnbr.c	
Funciones autorizadas: write	


- Escribe 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>
	Archivos a entregar: ft_atoi.c
	Funciones autorizadas: Ninguna

- Escribe una función que convierta el principio del string apuntado por str en un entero de tipo int
- str puede empezar con un número arbitrario de espacios (tal y como lo define `isspace(3)`)
- str puede ir seguido 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.
- str puede ir seguido de cualquier cantidad de número de dígitos en base 10
- Tu función tendrá que leer los caracteres de str, siempre que estos cumplan con las reglas mencionadas anteriormente, y tendrá que devolver el número encontrado hasta entonces.
- No deberías tener en cuenta los desbordamientos (overflows y underflows), en estos casos el resultado se considera indefinido.
- Puedes comparar tu función con la verdadera función atoi, quitando la parte de los signos y del overflow.
- Aquí tienes 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>
	Archivos a entregar: ft_putnbr_base.c
	Funciones autorizadas: write

- Escribe 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 un **string**.
- La base contiene todos los símbolos que se pueden utilizar para 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>
	Archivos a entregar: ft_atoi_base.c
	Funciones autorizadas: Ninguna

- Escribe 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);
```

Capítulo IX

Entrega y evaluación

Entrega tu proyecto en tu repositorio `Git` como de costumbre. Solo el trabajo entregado en el repositorio será evaluado durante la defensa. No dudes en comprobar varias veces los nombres de los archivos para verificar que sean correctos.



Sólo necesitas entregar los archivos requeridos por el enunciado de este proyecto.