



C Piscine

C 04

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

Versión: 5

Índice general

I.	Instrucciones	2
II.	Instrucciones sobre la IA	4
III.	Introducción	7
IV.	Ejercicio 00 : ft_strlen	8
V.	Ejercicio 01 : ft_putstr	9
VI.	Ejercicio 02 : ft_putnbr	10
VII.	Ejercicio 03 : ft_atoi	11
VIII.	Ejercicio 04 : ft_putnbr_base	12
IX.	Ejercicio 05 : ft_atoi_base	13
X.	Entrega y evaluación	14

Capítulo I

Instrucciones

- Esta página será la única referencia: no te fíes de los rumores.
- Antes de empezar a hacer los ejercicios deberás registrarte en el proyecto en tu intranet. Una vez lo hayas hecho, obtendrás tu repositorio Git en el cual debes trabajar.
- Los ejercicios han sido ordenados del más sencillo al más complejo. Por lo tanto, debes hacerlos en orden y asegurarte de que el ejercicio es correcto antes de continuar con el siguiente.
- Los ejercicios de Shell se deben ejecutar con `/bin/bash`.
- Los ejercicios de C se deben compilar con `cc` y utilizando las flags `-Wall -Wextra -Werror`.
- Solamente hay que entregar una función `main()` si lo que se pide es un programa. Si se pide una función se puede entregar el `main()` comentado con la batería de tests que hayas hecho.
- Los ejercicios de C se deben escribir de acuerdo a **la Norma**. Puedes encontrarla en la intranet, en el ícono de la brújula, FAQ, sección 42, apartado General Pedagogy como **La Norma**.
- Para ayudarte a comprobar si tus ejercicios cumplen **la Norma** puedes ayudarte de `norminette` - un programa que aplica ciertos requisitos de **la Norma** a tu código. Si tienes archivos o funciones adicionales, también deben cumplir con **la Norma**.
- Lee detenidamente los ejemplos. Podrían dar información que no se especifica en el enunciado...
- Asegúrate de que tus directorios y archivos tienen los permisos adecuados.
- Debes respetar **el procedimiento de entrega** para todos tus ejercicios. Solo el trabajo de tu repositorio Git será evaluado.
- **No puedes** dejar en tu directorio **ningún** archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.

- La evaluación de este proyecto consta de dos fases. Primero, las personas con las que compartes piscina se encargarán de evaluar tus ejercicios. A continuación, serán evaluados por un programa que se llama **la Moulinette**.
- Tus funciones no deben terminar de forma inesperada (segfault, bus error, double free) excepto en el caso de comportamientos indefinidos. Si esto sucede, tu proyecto será considerado no funcional y recibirás un 0 durante la evaluación.
- **La Moulinette** es muy estricta a la hora de evaluar y está completamente automatizada. Es imposible discutir con ella sobre tu nota. Por lo tanto, debes mantener el rigor en tu código para evitar cualquier sorpresa.
- **La Moulinette** utiliza **norminette** para comprobar tus ejercicios, aunque este programa no es exhaustivo en la verificación de **la Norma**. Durante las evaluaciones por parte de las personas de tu Piscina se comprobará que **la Norma** es correcta. Si se incumple el proyecto tendrá una puntuación final de 0, aunque **norminette** no indique ningún error.
- ¿Tienes alguna pregunta? Pregunta a la persona de tu derecha. Si no puede ayudarte, prueba con la persona de tu izquierda.
- ¡Por [Loki](#), por [Freyja](#)! ¡Piensa fuerte!!



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

Capítulo II

Instrucciones sobre la IA

Contexto

La Piscina de C es intensa. Es tu primer gran desafío en 42: una inmersión profunda en la resolución de problemas, la autonomía y la comunidad.

Durante esta etapa, tu objetivo principal es construir unas bases sólidas, a través del esfuerzo, la repetición y, sobre todo, mediante el **aprendizaje entre pares**.

En la era de la IA, los atajos son fáciles de encontrar. Sin embargo, es importante considerar si el uso que haces de la IA te está ayudando realmente a crecer, o si simplemente te está impidiendo desarrollar habilidades reales.

La Piscina también es una experiencia humana y, por ahora, nada puede reemplazar eso. Ni siquiera la IA.

Para obtener una visión más completa de nuestra postura sobre la IA como herramienta de aprendizaje, como parte del plan de estudios de TIC (Tecnologías de la Información y la Comunicación) y como una expectativa creciente en el mercado laboral, consulta las preguntas frecuentes *FAQ* disponibles en la intranet.

● Mensaje principal

- 👉 Construir fundamentos sólidos sin atajos.
- 👉 Desarrollar de forma real habilidades técnicas y transversales.
- 👉 Experimentar el aprendizaje entre pares de forma real, empezar a aprender a aprender y a resolver nuevos problemas.
- 👉 El proceso de aprendizaje es más importante que el resultado.
- 👉 Aprender sobre los riesgos asociados a la IA y desarrollar prácticas de control efectivas y medidas de mitigación para evitar errores comunes.

● Reglas para la piscina:

- Aplica la lógica y el razonamiento a las tareas asignadas, especialmente antes de recurrir a la IA.
- No deberías pedir respuestas directas a la IA.
- Aprende sobre el enfoque global de 42 sobre la IA.

● Resultados de esta etapa:

Durante esta etapa fundamental, obtendrás los siguientes resultados:

- Obtener Una base sólida técnica y en programación.
- Comprender por qué y cómo la IA puede ser peligrosa durante esta fase.

● Comentarios y ejemplos:

- Si, sabemos que la IA existe. Y sí, puede resolver tus proyectos. Pero estás aquí para aprender, no para demostrar que la IA ha aprendido. No pierdas tu tiempo (o el nuestro) solo para demostrar que la IA puede resolver el problema dado. y que puede resolver tus proyectos, pero estás aquí para aprender. Te recomendamos que inviertas 26 días de aprendizaje de calidad para resolver los retos que hemos pensado para ti.
- Aprender en 42 no va sólo de saber la respuesta. Se trata de desarrollar la capacidad de encontrar una. La IA te da la respuesta directamente, pero eso te impide construir tu propio razonamiento. Y el razonamiento requiere tiempo, esfuerzo y conlleva fracasos. El camino hacia el éxito no debe ser fácil. no tiene nada que ver con saber la respuesta. En 42 aprendemos desarrollando la capacidad para encontrarla. La IA te dará la respuesta directa, lo que impide que desarrolles tu propio razonamiento. Razonar requiere tiempo, esfuerzo y equivocarse muchas veces.
- Piensa que durante los exámenes, la IA no está disponible sin internet, sin smartphones, etc. Te darás cuenta rápidamente si has confiado demasiado en la IA durante tu proceso de aprendizaje. Recuerda que durante los exámenes no tendrás acceso a la IA ni a internet. Te vas a enfrentar a una hoja en blanco donde vas a tener que escribir tu propio código.
- El aprendizaje entre pares te expone a diferentes ideas y enfoques, mejorando tus habilidades interpersonales y tu capacidad de pensar de forma diferente. Eso es mucho más valioso que simplemente chatear con un bot. Así que, ¡que no te supere la timidez!: ¡habla, haz preguntas y aprende con el resto de la piscina!

- Sí, la IA formará parte del plan de estudios, tanto como herramienta de aprendizaje como tema en sí mismo. Incluso tendrás la oportunidad de crear tu propio software de IA. Para aprender más sobre nuestro enfoque progresivo, puedes consultar la documentación disponible en la intranet.

✓ Buenas prácticas:

Me atasco en un nuevo concepto. Le pregunto a alguien cercano cómo lo ha abordado. Hablamos durante 10 minutos y, de repente, todo encaja. Lo entiendo. No entiendo algo concreto del proyecto y no sé cómo continuar. Le pregunto a otra persona de la piscina cómo lo ha abordado, hablamos sobre el tema y, si es necesario, incluso utilizamos otros métodos (papel y boli, dibujos, metáforas, etc.) hasta conseguir entenderlo.

✗ Mala práctica:

Utilizo la IA en secreto, copio un código que parece correcto. Durante la evaluación entre pares, no puedo explicar nada. Suspendo. Durante el examen, sin IA, me vuelvo a atascar. Suspendo.

Capítulo III

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 IV

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 V

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 VI

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 VII

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 números 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 VIII

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 IX

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 X

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.