



C Piscine

C 05

Resumen: Este documento corresponde al enunciado del módulo C 05 de la C Piscine
de 42.

Versión: 7

Índice general

I.	Instrucciones	2
II.	Instrucciones sobre la IA	4
III.	Introducción	7
IV.	Ejercicio 00 : ft_iterative_factorial	9
V.	Ejercicio 01 : ft_recursive_factorial	10
VI.	Ejercicio 02 : ft_iterative_power	11
VII.	Ejercicio 03 : ft_recursive_power	12
VIII.	Ejercicio 04 : ft_fibonacci	13
IX.	Ejercicio 05 : ft_sqrt	14
X.	Ejercicio 06 : ft_is_prime	15
XI.	Ejercicio 07 : ft_find_next_prime	16
XII.	Ejercicio 08 : Las diez damas	17
XIII.	Entrega y evaluación	18

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 un texto extraído del primer libro de la saga de Harry Potter:

Oh, podrás pensar que no soy bonito,
pero no juzgues por lo que ves.
Me comeré a mí mismo si puedes encontrar
un sombrero más inteligente que yo.

Puedes tener bombines negros,
sombreros altos y elegantes.
Pero yo soy el Sombrero Seleccionador de Hogwarts
y puedo superar a todos.

No hay nada escondido en tu cabeza
que el Sombrero Seleccionador no pueda ver.
Así que pruébame y te diré
dónde debes estar.

Puedes pertenecer a Gryffindor,
donde habitan los valientes.
Su osadía, temple y caballerosidad
ponen aparte a los de Gryffindor.

Puedes pertenecer a Hufflepuff
donde son justos y leales.
Esos perseverantes Hufflepuff
de verdad no temen el trabajo pesado.

O tal vez a la antigua sabiduría de Ravenclaw,
Si tienes una mente dispuesta,
porque los de inteligencia y erudición
siempre encontrarán allí a sus semejantes.

O tal vez en Slytherin
harás tus verdaderos amigos.
Esa gente astuta utiliza cualquier medio
para lograr sus fines.

¡Así que pruébame! ¡No tengas miedo!
¡Y no recibirás una bofetada!
Estás en buenas manos (aunque yo no las tenga).
Porque soy el Sombrero Pensante.

Desgraciadamente, este tema no tiene nada que ver con la saga de Harry Potter y es una lástima, porque el proyecto no se va a realizar por arte de magia.

Capítulo IV

Ejercicio 00 : ft_iterative_factorial

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

- Escribe una función iterativa que devuelva un número. Este número será el resultado de la operación factorial a partir del número usado como parámetro.
- Si el argumento no es válido, la función debe devolver 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- El prototipo de la función deberá ser el siguiente:

```
int ft_iterative_factorial(int nb);
```

Capítulo V

Ejercicio 01 : ft_recursive_factorial

	Ejercicio: 01
	ft_recursive_factorial
	Directorio de entrega: <i>ex01/</i>
	Archivos a entregar: ft_recursive_factorial.c
	Funciones autorizadas: Ninguna

- Escribe una función recursiva que devuelva el factorial de un número usado como parámetro.
- Si el argumento no es válido, la función debe devolver 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- El prototipo de la función deberá ser el siguiente:

```
int ft_recursive_factorial(int nb);
```

Capítulo VI

Ejercicio 02 : ft_iterative_power

	Ejercicio: 02
	ft_iterative_power
	Directorio de entrega: <i>ex02/</i>
	Archivos a entregar: ft_iterative_power.c
	Funciones autorizadas: Ninguna

- Escribe una función iterativa que devuelva una potencia de un número.
- Una potencia inferior a 0 devolverá 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- Se ha decidido que 0 potencia 0 devolverá 1
- El prototipo de la función deberá ser el siguiente:

```
int ft_iterative_power(int nb, int power);
```

Capítulo VII

Ejercicio 03 : ft_recursive_power

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

- Escribe una función recursiva que devuelva una potencia de un número.
- Una potencia inferior a 0 devolverá 0.
- No hay que gestionar los “int overflow”, el retorno de la función será indefinido.
- Hemos decidido que 0 potencia 0 devolverá 1
- El prototipo de la función deberá ser el siguiente:

```
int ft_recursive_power(int nb, int power);
```

Capítulo VIII

Ejercicio 04 : ft_fibonacci

	Ejercicio: 04
	ft_fibonacci
	Directorio de entrega: <i>ex04/</i>
	Archivos a entregar: ft_fibonacci.c
	Funciones autorizadas: Ninguna

- Escribe una función **ft_fibonacci** que devuelva el n -ésimo elemento de la sucesión de Fibonacci; El primer elemento de la serie está en el índice 0. Consideraremos que la sucesión de Fibonacci comienza por 0, 1, 1, 2.
- No se tendrán que gestionar los overflows.
- El prototipo de la función deberá ser el siguiente:

```
int ft_fibonacci(int index);
```

- Por supuesto, **ft_fibonacci** tendrá que ser recursiva.
- Si el **índice** es inferior a 0, la función retornará -1.

Capítulo IX

Ejercicio 05 : ft_sqrt

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

- Escribe una función que devuelva la raíz cuadrada entera de un número, si existe, o 0 si la raíz cuadrada no es entera.
- El prototipo de la función deberá ser el siguiente:

```
int ft_sqrt(int nb);
```

Capítulo X

Ejercicio 06 : ft_is_prime

	Ejercicio: 06
	ft_is_prime
	Directorio de entrega: <i>ex06/</i>
	Archivos a entregar: ft_is_prime.c
	Funciones autorizadas: Ninguna

- Escribe una función que devuelva 1 si el número es primo y 0 si el número no lo es.
- El prototipo de la función deberá ser el siguiente:

```
int ft_is_prime(int nb);
```



0 y 1 no son números primos.

Capítulo XI

Ejercicio 07 : ft_find_next_prime

	Ejercicio: 07
	ft_find_next_prime
	Directorio de entrega: <i>ex07/</i>
	Archivos a entregar: ft_find_next_prime.c
	Funciones autorizadas: Ninguna

- Escribe una función que devuelva el número primo inmediatamente superior o igual al número usado como parámetro.
- El prototipo de la función deberá ser el siguiente:

```
int ft_find_next_prime(int nb);
```

Capítulo XII

Ejercicio 08 : Las diez damas

	Ejercicio: 08
	Las diez damas
	Directorio de entrega: <i>ex08/</i>
	Archivos a entregar: ft_ten_queens_puzzle.c
	Funciones autorizadas: write

- Escribe una función que muestre todas las posibilidades de colocar diez damas en un tablero de 10x10 sin que se puedan alcanzar con una sola jugada y que devuelva el número de posibilidades.
- Se tendrá que utilizar la recursividad.
- El valor retornado por su función tendrá que ser el número de soluciones mostradas.
- El prototipo de la función deberá ser el siguiente:

```
int ft_ten_queens_puzzle(void);
```

- La visualización se hará de la manera siguiente:

```
$>./a.out | cat -e
0257948136$
0258693147$
...
4605713829$
4609582731$
...
9742051863$
$>
```

- La sucesión se lee de izquierda a derecha. La primera cifra corresponde a la posición de la primera dama en la primera columna (el índice empieza con 0). La n-ésima cifra corresponde a la posición de la n-ésima dama en la n-ésima columna.

Capítulo XIII

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.