



C Piscine

C 02

Resumen: Este documento corresponde al enunciado del proyecto C 02 de la C Piscine de 42.

Versión: 6

Índice general

I.	Instrucciones	2
II.	Instrucciones sobre la IA	4
III.	Introducción	7
IV.	Ejercicio 00 : <code>ft_strcpy</code>	9
V.	Ejercicio 01 : <code>ft_strncpy</code>	10
VI.	Ejercicio 02 : <code>ft_str_is_alpha</code>	11
VII.	Ejercicio 03 : <code>ft_str_is_numeric</code>	12
VIII.	Ejercicio 04 : <code>ft_str_is_lowercase</code>	13
IX.	Ejercicio 05 : <code>ft_str_is_uppercase</code>	14
X.	Ejercicio 06 : <code>ft_str_is_printable</code>	15
XI.	Ejercicio 07 : <code>ft_strupcase</code>	16
XII.	Ejercicio 08 : <code>ft_strlowcase</code>	17
XIII.	Ejercicio 09 : <code>ft_strcapitalize</code>	18
XIV.	Ejercicio 10 : <code>ft_strlcpy</code>	19
XV.	Ejercicio 11 : <code>ft_putstr_non_printable</code>	20
XVI.	Ejercicio 12 : <code>ft_print_memory</code>	21
XVII.	Entrega y evaluación	23

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 icono 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. Suspenso. Durante el examen, sin IA, me vuelvo a atascar. Suspenso.

Capítulo III

Introducción

He aquí un diálogo extraído de la serie Silicon Valley:

- I mean, why not just use Vim over Emacs? (CHUCKLES)
- I do use Vim over Emacs.
- Oh, God, help us! Okay, uh you know what? I just don't think this is going to work. I'm so sorry. Uh, I mean like, what, we're going to bring kids into this world with that over their heads? That's not really fair to them, don't you think?
- Kids? We haven't even slept together.
- And guess what, it's never going to happen now, because there is no way I'm going to be with someone who uses spaces over tabs.
- Richard! (PRESS SPACE BAR MANY TIMES)
- Wow. Okay. Goodbye.
- One tab saves you eight spaces! - (DOOR SLAMS) - (BANGING)

. . .

(RICHARD MOANS)

- Oh, my God! Richard, what happened?
- I just tried to go down the stairs eight steps at a time. I'm okay, though.
- See you around, Richard.
- Just making a point.

Afortunadamente, no estás obligado a utilizar emacs y su barra espaciadora para completar los ejercicios siguientes.


La meta de hoy

La puntuación mínima para validar este proyecto es del 50 %.

Depende de ti decidir qué ejercicios te permitirán alcanzar esta puntuación y si quieres hacer más o no.

Capítulo IV

Ejercicio 00 : ft_strcpy


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

- Reproduce el comportamiento de la función **strcpy** (man strcpy)
- El prototipo de la función deberá ser el siguiente:

```
char *ft_strcpy(char *dest, char *src);
```

Capítulo V

Ejercicio 01 : ft_strncpy


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

- Reproduce el comportamiento de la función **strncpy** (man strncpy)
- El prototipo de la función deberá ser el siguiente:

```
char      *ft_strncpy(char *dest, char *src, unsigned int n);
```

Capítulo VI

Ejercicio 02 : ft_str_is_alpha

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


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

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

- Tendrá que devolver 1 si **str** es un string vacío.

Capítulo VII

Ejercicio 03 : ft_str_is_numeric

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


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente dígitos y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

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

- Tendrá que devolver 1 si **str** es un string vacío.

Capítulo VIII

Ejercicio 04 : ft_str_is_lowercase

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


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos en minúsculas y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

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

- Tendrá que devolver 1 si **str** es un string vacío.

Capítulo IX

Ejercicio 05 : ft_str_is_uppercase

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


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos en mayúsculas y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

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

- Tendrá que devolver 1 si **str** es un string vacío.

Capítulo X

Ejercicio 06 : ft_str_is_printable

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


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres imprimibles y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

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

- Tendrá que devolver 1 si **str** es un string vacío.

Capítulo XI

Ejercicio 07 : ft_strupcase

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


- Crea una función que ponga cada letra en mayúscula.
- El prototipo de la función deberá ser el siguiente:

```
char *ft_strupcase(char *str);
```

- Deberá devolver **str**.

Capítulo XII

Ejercicio 08 : ft_strlowcase

	Ejercicio: 08
	ft_strlowcase
	Directorio de entrega: <i>ex08/</i>
	Archivos a entregar: ft_strlowcase.c
	Funciones autorizadas: Ninguna

- Crea una función que ponga cada letra en minúscula.
- El prototipo de la función deberá ser el siguiente:

```
char *ft_strlowcase(char *str);
```

- Deberá devolver **str**.

Capítulo XIII

Ejercicio 09 : ft_strcapitalize

	Ejercicio: 09
ft_strcapitalize	
Directorio de entrega: <i>ex09/</i>	
Archivos a entregar: ft_strcapitalize.c	
Funciones autorizadas: Ninguna	

- Creaa una función que ponga en mayúscula la primera letra de cada palabra y el resto de la palabra en minúsculas.
- Una palabra es una secuencia (string) de caracteres alfanuméricos.
- El prototipo de la función deberá ser el siguiente:

```
char      *ft_strcapitalize(char *str);
```

- Deberá devolver **str**.
- Por ejemplo:


```
salut, comment tu vas ? 42mots quarante-deux; cinquante+et+un
```

- Se convierte en:

```
Salut, Comment Tu Vas ? 42mots Quarante-Deux; Cinquante+Et+Un
```

Capítulo XIV

Ejercicio 10 : ft_strlcpy


	Ejercicio: 10
	ft_strlcpy
	Directorio de entrega: <i>ex10/</i>
	Archivos a entregar: ft_strlcpy.c
	Funciones autorizadas: Ninguna

- Reproduce el comportamiento de la función `strlcpy` (man `strlcpy`)
- El prototipo de la función deberá ser el siguiente:

```
unsigned int ft_strlcpy(char *dest, char *src, unsigned int size);
```

Capítulo XV

Ejercicio 11 : ft_putstr_non_printable

	Ejercicio: 11
	ft_putstr_non_printable
	Directorio de entrega: <i>ex11/</i>
	Archivos a entregar: ft_putstr_non_printable.c
	Funciones autorizadas: write

- Escribe una función que muestre una secuencia de caracteres en la pantalla. Si esta secuencia contiene caracteres no imprimibles, deberán ser mostrados en formato hexadecimal (en minúsculas) precedidos de una barra invertida (backslash).
- Por ejemplo, con este parámetro:

```
Coucou\ntu vas bien ?
```

- La función tendrá que mostrar:


```
Coucou\0atu vas bien ?
```

- El prototipo de la función deberá ser el siguiente:

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

Capítulo XVI

Ejercicio 12 : `ft_print_memory`

	Ejercicio: 12
	<code>ft_print_memory</code>
	Directorio de entrega: <i>ex12/</i>
	Archivos a entregar: <code>ft_print_memory.c</code>
	Funciones autorizadas: <code>write</code>

- Escriba una función que muestre la región de memoria en la pantalla.
- La visualización de la región de memoria estará dividida en tres “columnas” separadas por un espacio:
 - La dirección en hexadecimal del primer carácter de la línea seguido de “:”.
 - El contenido en hexadecimal, con un espacio cada dos caracteres, deberá ser completado con espacios si es preciso (ver el ejemplo a continuación).
 - El contenido en caracteres imprimibles.
- Si un carácter es no imprimible será remplazado por un punto.
- Cada línea debe contener dieciséis caracteres.
- Si `size` es igual a 0, no se muestra nada.

- Ejemplo:

```
$> ./ft_print_memory
000000010a161f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin
000000010a161f50: 6368 6573 090a 0963 0720 6573 7420 666f ches...c. est fo
000000010a161f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on
000000010a161f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.
000000010a161f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..
000000010a161f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000 ..lol.lol. .
$> ./ft_print_memory | cat -te
0000000107ff9f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin$
0000000107ff9f50: 6368 6573 090a 0963 0720 6573 7420 666f ches...c. est fo$
0000000107ff9f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on $
0000000107ff9f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.$
0000000107ff9f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..$
0000000107ff9f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000 ..lol.lol. .$.
$>
```

- El prototipo de la función deberá ser el siguiente:

```
void      *ft_print_memory(void *addr, unsigned int size);
```

- Deberá devolver addr.

Capítulo XVII

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.