



C Piscine

C 07

Resumen: Este documento corresponde al enunciado del módulo C 07 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 : ft_strdup	8
V.	Ejercicio 01 : ft_range	9
VI.	Ejercicio 02 : ft_ultimate_range	10
VII.	Ejercicio 03 : ft_strjoin	11
VIII.	Ejercicio 04 : ft_convert_base	12
IX.	Ejercicio 05 : ft_split	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 una lista de los monstruos que se pueden encontrar en las famosas mazmorras de Naheulbeuk:

- Todo tipo de zombis;
- Arañas gigantes;
- Orcos;
- Goblins;
- Trolls de las cavernas;
- Brujos;
- Guerreros malditos;
- Ratas mutantes;
- Una botella de aceite;
- Papel higiénico;
- Dos esponjas;
- Raviolis.

Capítulo IV

Ejercicio 00 : ft_strdup

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

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

```
char *ft_strdup(char *src);
```

Capítulo V

Ejercicio 01 : ft_range

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

- Escribe una función `ft_range` que devuelva un array de `int`. Este array de `int` contendrá todos los valores entre `min` y `max`.
- `Min` incluido - `max` excluido.
- El prototipo de la función deberá ser el siguiente:

```
int *ft_range(int min, int max);
```

- Si el valor `min` es superior o igual al valor `max`, se devolverá un puntero nulo.

Capítulo VI

Ejercicio 02 : ft_ultimate_range

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

- Escribe una función **ft_ultimate_range** que asigne memoria y datos a un array de **int**. Este array de **int** contendrá todos los valores entre **min** y **max**.
- **Min** incluido - **max** excluido.
- El prototipo de la función deberá ser el siguiente:

```
int      ft_ultimate_range(int **range, int min, int max);
```

- Devolverá el tamaño de **range** (o -1 en caso de problemas).
- Si el valor **min** es superior o igual al valor **max**, **range** apuntará a NULL y retornará 0.

Capítulo VII

Ejercicio 03 : ft_strjoin

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

- Escribe una función que concatene el conjunto de cadenas de caracteres apuntadas por **strs**, separándolas por **sep**.
- **size** representa el tamaño de **strs**.
- Si **size** vale 0, habrá que retornar una cadena de caracteres vacía que puedas liberar con **free()**.
- El prototipo de la función deberá ser el siguiente:

```
char *ft_strjoin(int size, char **strs, char *sep);
```

Capítulo VIII

Ejercicio 04 : ft_convert_base

	Ejercicio: 04
	ft_convert_base
	Directorio de entrega: <i>ex04/</i>
	Archivos a entregar: ft_convert_base.c , ft_convert_base2.c
	Funciones autorizadas: malloc , free

- Escribe una función que devuelva el resultado de la conversión de la cadena **nbr**, expresada en una base **base_from**, a una base **base_to**.
- **nbr**, **base_from**, **base_to** no son necesariamente modificables.
- **nbr** seguirá las mismas reglas que **ft_atoi_base**. Por lo tanto, cuidado con el + , - y con los espacios blancos.
- El número representado por **nbr** cabe en un **int**.
- Si una base es incorrecta, la función retornará NULL.
- Si es preciso que el número devuelto lleve un prefijo, este será un solo y único - , sin espacios ni + .
- El prototipo de la función deberá ser el siguiente:

```
char *ft_convert_base(char *nbr, char *base_from, char *base_to);
```

Capítulo IX

Ejercicio 05 : ft_split

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

- Escribe una función que divida una cadena de caracteres en función de otra cadena de caracteres.
- Habrá que utilizar cada carácter de la cadena **charset** como separador.
- La función retorna un array donde cada uno de sus elementos contiene la dirección de una cadena de caracteres comprendida entre dos separadores. El último elemento del array tendrá que ser igual a 0 para indicar el final del array.
- Tu array no puede tener cadenas vacías. Saca las conclusiones pertinentes.
- La cadena pasada como argumento no será modificable.
- El prototipo de la función deberá ser el siguiente:

```
char **ft_split(char *str, char *charset);
```

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.