



# Piscina C

## C 09

*Sumário: Este documento é o tema do módulo C 09 da Piscina C da 42.*

*Versão: 5*

# Conteúdo

<b>I</b>	<b>Instruções</b>	<b>2</b>
<b>II</b>	<b>Intruções IA</b>	<b>4</b>
<b>III</b>	<b>Preâmbulo</b>	<b>7</b>
<b>IV</b>	<b>Exercício 00 : libft</b>	<b>8</b>
<b>V</b>	<b>Exercício 01 : Makefile</b>	<b>9</b>
<b>VI</b>	<b>Exercise 02 : ft_split</b>	<b>11</b>
<b>VII</b>	<b>Submission and peer-evaluation</b>	<b>12</b>

# Capítulo I

## Instruções

- Somente este documento servirá de referência; não confie nos boatos.
- Leia bem o enunciado antes de entregar os seus exercícios. A qualquer momento pode haver alterações.
- Tenha atenção aos direitos dos seus ficheiros e pastas.
- Deverá seguir o procedimento de entrega para todos os exercícios.
- Os seus exercícios serão corrigidos pelos seus colegas de piscine.
- Além dos seus colegas, a Moulinette também corrigirá os seus exercícios.
- A Moulinette é extremamente rígida na sua avaliação. É completamente automatizada, e é impossível discutir a sua nota com ela. Portanto, seja rigoroso!
- A Moulinette não tem uma mente muito aberta: não tenta entender código que não respeita a Norma. A Moulinette utiliza o programa `norminette` para verificar a norma dos ficheiros. Seria uma tontice entregar código que não passa pela `norminette`...
- Os exercícios são ordenados precisamente do mais simples ao mais complexo. Em caso algum consideraremos um exercício mais complexo se outro mais simples não tiver sido perfeitamente realizado.
- A utilização de qualquer função proibida é um caso de fraude. Qualquer fraude é punida com nota de -42.
- Deve entregar uma função `main()` se for pedido um programa.
- A Moulinette compila com as flags `-Wall -Wextra -Werror`, e utiliza `cc`.
- Se o seu programa não compila, terá 0.
- Não deve deixar no repositório de entrega nenhum outro ficheiro além daqueles explicitamente especificados pelo enunciado dos exercícios.

- Tem alguma dúvida? Pergunte ao seu vizinho da direita. Tente, também, com o seu vizinho da esquerda.
- A bibliografia para consulta chama-se `Google / man / Internet / ....`
- Considere discutir os exercícios no Slack da sua piscine!
- Leia atentamente os exemplos: podem demonstrar coisas que não estão especificadas no enunciado...



A Norminette deve ser lançada com a *flag* `-R CheckForbiddenSourceHeader`. A Moulinette também a utilizará.

# Capítulo II

## Intruições IA

### Contexto

A Piscina C é intensa. É o teu primeiro grande desafio na 42 — um mergulho profundo na resolução de problemas, autonomia e comunidade.

Nesta fase, o teu principal objetivo é obter uma base sólida — através do esforço, da repetição e, acima de tudo, da partilha de aprendizagens com os teus colegas.

Na era da IA, os atalhos são fáceis de encontrar. No entanto, é importante considerar se o uso da IA está realmente a ajudar-te a crescer — ou apenas a impedir-te de desenvolver competências reais.

A Piscine também é uma experiência humana — e, por agora, nada substitui isso. Nem mesmo a IA.

Para uma visão mais completa da nossa posição sobre a IA — como ferramenta de aprendizagem, como parte do currículo TIC e como uma expectativa crescente no mercado de trabalho — consulta o FAQ dedicado disponível no intranet.

### ● Mensagem principal

- ✎ Constrói bases sólidas sem atalhos.
- ✎ Desenvolve verdadeiramente competências técnicas e interpessoais.
- ✎ Vive a aprendizagem entre pares, começa a aprender a aprender e a resolver novos problemas.
- ✎ A jornada de aprendizagem é mais importante do que o resultado.
- ✎ Aprende os riscos associados à IA e desenvolve práticas de controlo eficazes e contramedidas para evitar os erros mais comuns.

## ● Regras para os alunos:

- Deves aplicar o raciocínio nas tarefas atribuídas, especialmente antes de recorrereres à IA.
- Não deves pedir respostas diretas à IA.
- Deves aprender sobre a abordagem global da 42 em relação à IA.

## ● Resultados esperados:

Nesta fase, vais ter os seguintes resultados:

- Obter bases sólidas em tecnologia e programação.
- Compreender por que razão e de que forma a IA pode ser perigosa durante esta fase.

## ● Comentários e exemplos:

- Sim, sabemos que a IA existe — e sim, pode resolver os teus projetos. Mas estás aqui para aprender, não para provar que a IA já aprendeu. Não percas tempo (nem o nosso) apenas para demonstrar que a IA consegue resolver o problema.
- Aprender na 42 não é sobre saber a resposta — é sobre desenvolver a capacidade de encontrar uma. A IA dá-te a resposta diretamente, mas isso impede-te de construir o teu próprio raciocínio. E o raciocínio exige tempo, esforço e envolve falhas. O caminho para o sucesso não deve ser fácil.
- Lembra-te que nos exames a IA não está disponível — sem internet, sem telemóveis, etc. Vais perceber rapidamente se dependeste demasiado da IA no teu processo de aprendizagem.
- A aprendizagem entre pares expõe-te a ideias e abordagens diferentes, melhorando as tuas competências interpessoais e a tua capacidade de pensar de forma divergente. Isso é muito mais valioso do que conversar com um bot. Por isso, não sejas tímido — fala, faz perguntas e aprende em conjunto!
- Sim, a IA fará parte do currículo — tanto como ferramenta de aprendizagem como tema de estudo. Terás até a oportunidade de construir o teu próprio software de IA. Para saberes mais sobre a nossa abordagem em crescendo, consulta a documentação disponível no intranet.

**✓ Boa prática:**

Estou com dificuldades num novo conceito. Pergunto a alguém ao meu lado como o abordou. Falamos durante 10 minutos — e de repente faz sentido. Percebo.

**✗ Má prática:**

Uso a IA em segredo, copio algum código que parece estar certo. Durante a avaliação por pares, não consigo explicar nada. Falho. Durante o exame — sem IA — fico novamente bloqueado. Falho.

# Capítulo III

## Preâmbulo

Dialog from the movie The Big Lebowski:

The Dude: Walter, ya know, it's Smokey, so his toe slipped over the line a little, big deal. It's just a game, man.

Walter Sobchak: Dude, this is a league game, this determines who enters the next round robin. Am I wrong? Am I wrong?

Smokey: Yeah, but I wasn't over. Gimme the marker Dude, I'm marking it 8.

Walter Sobchak: [pulls out a gun] Smokey, my friend, you are entering a world of pain.

The Dude: Walter...

Walter Sobchak: You mark that frame an 8, and you're entering a world of pain.

Smokey: I'm not...

Walter Sobchak: A world of pain.

Smokey: Dude, he's your partner...

Walter Sobchak: [shouting] Has the whole world gone crazy? Am I the only one around here who gives a shit about the rules? Mark it zero!

The Dude: They're calling the cops, put the piece away.

Walter Sobchak: Mark it zero!

[points gun in Smokey's face]

The Dude: Walter...

Walter Sobchak: [shouting] You think I'm fucking around here? Mark it zero!


Smokey: All right, it's fucking zero. Are you happy, you crazy fuck?

Walter Sobchak: ...It's a league game, Smokey.



# Capítulo IV

## Exercício 00 : libft

	Exercício : 00
libft	
Pasta de entrega : <i>ex00/</i>	
Ficheiros para entregar : <code>libft_creator.sh</code> , <code>ft_putchar.c</code> , <code>ft_swap.c</code> , <code>ft_putstr.c</code> , <code>ft_strlen.c</code> , <code>ft_strcmp.c</code>	
Funções autorizadas : <code>write</code>	

- Crie a sua biblioteca `ft`. Ela vai chamar-se `libft.a`.
- Um script shell chamado `libft_creator.sh` vai compilar adequadamente os ficheiros de código-fonte e criar a sua biblioteca.
- Essa biblioteca deve conter todas as seguintes funções:


```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- Nós vamos executar o seguinte comando:

```
sh libft_creator.sh
```

# Capítulo V

## Exercício 01 : Makefile

	Exercício : 01
Makefile	
Pasta de entrega : <i>ex01/</i>	
Ficheiros para entregar : <b>Makefile</b>	
Funções autorizadas : <b>Nenhuma</b>	

- Escreva o **Makefile** que compile a biblioteca **libft.a**.
- O seu **makefile** deve mostrar claramente cada comando que efetua.
- O seu **makefile** não deve efetuar comandos inúteis.
- O **Makefile** vai procurar os ficheiros de código-fonte na pasta "srcs".
- Esses ficheiros de código-fonte serão: **ft\_putchar.c**, **ft\_swap.c**, **ft\_putstr.c**, **ft\_strlen.c**, **ft\_strcmp.c**
- O **Makefile** vai procurar os ficheiros de cabeçalhos na pasta "includes".
- Esses ficheiros de cabeçalhos serão: **ft.h**
- Ele deverá compilar seus ficheiros c utilizando **cc** e as opções de sinalizações **-Wall** **-Wextra** **-Werror** nessa ordem.
- A lib estará na origem do exercício.
- Os ficheiros **.o** deverão estar ao lado do seu respectivo arquivo **.c**.
- O **Makefile** deverá também implementar as regras **clean**, **fclean**, **re**, **all** e, claro, **libft.a**.
- Executar apenas **make** deve ser equivalente a **make all**.


- A regra `all` deverá agir como `make libft.a`.
- A regra `clean` deverá eliminar todos os ficheiros temporários gerados.
- A regra `fclean` é o equivalente a um `make clean` e também apaga os binários criado no `make all`.
- A regra `re` é o equivalente a um `make fclean` seguido de um `make all`.
- Seu makefile não deve recompilar os ficheiros desnecessariamente.
- Nós vamos pegar só no seu Makefile e testar com os nossos ficheiros.



Cuidado com os wildcards!

# Capítulo VI

## Exercise 02 : ft\_split

	Exercício : 02
ft_split	
Pasta de entrega : <i>ex02/</i>	
Ficheiros para entregar : <code>ft_split.c</code>	
Funções autorizadas : <code>malloc</code>	

- Create a function that splits a string, using each character from the `charset` string as a separator.
- You will need to use each character from the `charset` string individually as a separator.
- The function should return an array where each element contains the address of a substring wrapped between two separators. The last element of the array should be NULL to indicate the end of the array.
- There should be no empty strings in your array. Draw your conclusions accordingly.
- The string provided as an argument cannot be modified.
- Here's the function prototype:

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

# Capítulo VII

## Submission and peer-evaluation

Submit your assignment to your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Make sure to double-check the filenames to ensure they are correct.



You must submit only the files required by the project instructions.