



# Piscina C

## C 11

*Sumário: Este documento é o tema do módulo C 11 da Piscina C da 42.*

*Versão: 8*

# 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 : ft_foreach</b>	<b>9</b>
<b>V</b>	<b>Exercício 01 : ft_map</b>	<b>10</b>
<b>VI</b>	<b>Exercício 02 : ft_any</b>	<b>11</b>
<b>VII</b>	<b>Exercício 03 : ft_count_if</b>	<b>12</b>
<b>VIII</b>	<b>Exercício 04 : ft_is_sort</b>	<b>13</b>
<b>IX</b>	<b>Exercício 05 : do-op</b>	<b>14</b>
<b>X</b>	<b>Exercício 06 : ft_sort_string_tab</b>	<b>17</b>
<b>XI</b>	<b>Exercício 07 : ft_advanced_sort_string_tab</b>	<b>18</b>
<b>XII</b>	<b>Submissão e avaliação</b>	<b>19</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

Here's a little story :

(1982, California) Larry Walters of Los Angeles is one of the few to contend for the Darwin Awards and live to tell the tale. "I have fulfilled my 20-year dream," said Walters, a former truck driver for a company that makes TV commercials. "I'm staying on the ground. I've proved the thing works."

Larry's boyhood dream was to fly. But fates conspired to keep him from his dream. He joined the Air Force, but his poor eyesight disqualified him from the job of pilot. After he was discharged from the military, he sat in his backyard watching jets fly overhead.

He hatched his weather balloon scheme while sitting outside in his "extremely comfortable" Sears lawnchair. He purchased 45 weather balloons from an Army-Navy surplus store, tied them to his tethered lawnchair (dubbed the Inspiration I) and filled the four-foot diameter balloons with helium. Then, armed with some sandwiches, Miller Lite, and a pellet gun, he strapped himself into his lawnchair. He figured he would shoot to pop a few of the many balloons when it was time to descend.

Larry planned to sever the anchor and lazily float to a height of about 30 feet above the backyard, where he would enjoy a few hours of flight before coming back down. But things didn't work out quite as Larry planned.

When his friends cut the cord anchoring the lawnchair to his Jeep, he did not float lazily up to 30 feet. Instead he streaked into the LA sky as if shot from a cannon, pulled by the lift of 45 helium balloons, holding 33 cubic feet of helium each.

He didn't level off at 100 feet, nor did he level off at 1000 feet. After climbing and climbing, he leveled off at 16,000 feet.

At that height he felt he couldn't risk shooting any of the balloons, lest he unbalance the load and really find himself in trouble. So he stayed there, drifting cold and frightened with his beer and sandwiches, for more than 14 hours. He crossed the primary approach corridor of LAX, where startled Trans World Airlines and Delta Airlines pilots radioed in reports



of the strange sight.


Eventually he gathered the nerve to shoot a few balloons, and slowly descended. The hanging tethers tangled and caught in a power line, blacking out a Long Beach neighborhood for 20 minutes. Larry climbed to safety, where he was arrested by waiting members of the LAPD. As he was led away in handcuffs, a reporter dispatched to cover the daring rescue asked him why he had done it. Larry replied nonchalantly, "A man can't just sit around."

The Federal Aviation Administration was not amused. Safety Inspector Neal Savoy said, "We know he broke some part of the Federal Aviation Act, and as soon as we decide which part it is, a charge will be filed."

A moral da história é que o Larry Walters deveria ter ficado na sua cadeira e aprendido C....

# Capítulo IV

## Exercício 00 : ft\_foreach

	Exercício : 00
	ft_foreach
	Pasta de entrega : <i>ex00/</i>
	Ficheiros para entregar : <b>ft_foreach.c</b>
	Funções autorizadas : <b>Nenhuma</b>

- Escreve uma função `ft_foreach` que, para uma array de inteiros, aplique uma função sobre todos os elementos do array. Essa função será aplicada na ordem do array.
- A função deve ser prototipada da seguinte maneira:


```
void      ft_foreach(int *tab, int length, void(*f)(int));
```

- Por exemplo, a função `ft_foreach` poderá ser chamada da seguinte forma para mostrar o conjunto de inteiros do array:

```
ft_foreach(tab, 1337, &ft_putnbr);
```

# Capítulo V

## Exercício 01 : ft\_map


	Exercício : 01
ft_map	
Pasta de entrega : <i>ex01/</i>	
Ficheiros para entregar : <b>ft_map.c</b>	
Funções autorizadas : <b>malloc</b>	

- Escreve uma função **ft\_map** que, para uma array de inteiros, aplicará uma função sobre todos os elementos do array (em sequência) e retornará um array de todos os valores retornados.
- A função será aplicada na ordem do array.
- A função deve ser prototipada da seguinte maneira:

```
int      *ft_map(int *tab, int length, int(*f)(int));
```

# Capítulo VI

## Exercício 02 : ft\_any

	Exercício : 02
ft_any	
Pasta de entrega : <i>ex02/</i>	
Ficheiros para entregar : <b>ft_any.c</b>	
Funções autorizadas : Nenhuma	


- Escreve uma função **ft\_any** que retornará 1 se pelo menos um elemento do array retornar algo diferente de 0 ao ser passado para a função **f**. Caso contrário, retorna 0.
- A função será aplicada na ordem do array.
- A função deve ser prototipada da seguinte maneira:

```
int ft_any(char **tab, int(*f)(char*));
```

- O array termina em nulo.

# Capítulo VII

## Exercício 03 : ft\_count\_if


	Exercício : 03
	ft_count_if
	Pasta de entrega : <i>ex03/</i>
	Ficheiros para entregar : <code>ft_count_if.c</code>
	Funções autorizadas : Nenhuma

- Escreve uma função `ft_count_if` que retornará o número de elementos do array que, ao serem passados para a função `f`, não retornam 0.
- A função será aplicada na ordem do array.
- A função deve ser prototipada da seguinte maneira:

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

# Capítulo VIII

## Exercício 04 : ft\_is\_sort


	Exercício : 04
	ft_is_sort
	Pasta de entrega : ex04/
	Ficheiros para entregar : ft_is_sort.c
	Funções autorizadas : Nenhuma

- Escreve uma função `ft_is_sort` que retornará 1 se o array estiver ordenado e 0 no caso contrário.
- A função passada como parâmetro retornará um inteiro negativo se o primeiro argumento for inferior ao segundo, 0 se forem iguais e um inteiro positivo caso contrário.
- A função deve ser prototipada da seguinte maneira:

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

# Capítulo IX

## Exercício 05 : do-op

	Exercício : 05
do-op	
Pasta de entrega : <i>ex05/</i>	
Ficheiros para entregar : Todos os arquivos necessários para o seu programa	
Funções autorizadas : <code>write</code>	

- Escreve um programa chamado `do-op`.
- O programa deve ser executado com três argumentos: `do-op valor1 operador valor2`
- Exemplo:

```
$> ./do-op 42 "+" 21
63
$>
```

- Deves utilizar um array de ponteiros para função de forma a chamar a função correspondente ao operador.
- Em caso de operador desconhecido, seu programa deverá mostrar 0.
- Se o número de argumentos não estiver correto, `do-op` não mostra nada.
- Seu programa deve aceitar e mostrar o resultado com os seguintes operadores: `'+', '-', '/', '*'` e `'%'`
- Your program should treat the values as integers.
- O seu programa deverá tratar valores como inteiros.
- Em caso de divisão por 0, seu programa deve mostrar:

```
Stop : division by zero
```

- Em caso de módulo por 0, seu programa deve mostrar:

```
Stop : modulo by zero
```




- Veja um exemplo de testes da Moulinette :

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

# Capítulo X

## Exercício 06 : ft\_sort\_string\_tab


	Exercício : 06
	ft_sort_string_tab
	Pasta de entrega : <i>ex06/</i>
	Ficheiros para entregar : <code>ft_sort_string_tab.c</code>
	Funções autorizadas : Nenhuma

- Escreve uma função `ft_sort_string_tab` que classifica por ordem `ascii` as `strings`.
- `tab` termina em nulo
- A ordenação será realizada ao trocar os ponteiros do array.
- Deve ser prototipada da seguinte maneira:

```
void ft_sort_string_tab(char **tab);
```

# Capítulo XI

## Exercício 07 : ft\_advanced\_sort\_string\_tab

	Exercício : 07
	ft_advanced_sort_string_tab
	Pasta de entrega : <i>ex07/</i>
	Ficheiros para entregar : <code>ft_advanced_sort_string_tab.c</code>
	Funções autorizadas : Nenhuma

- Escreve a função `ft_advanced_sort_string_tab` que ordena conforme o retorno da função passada como parâmetro
- A ordenação será realizada ao trocar os ponteiros da array.
- `tab` termina em nulo
- Deve ser prototipada da seguinte maneira:

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



Se chamares o `ft_advanced_sort_string_tab()` como segundo parâmetro `ft_strcmp` retornará o mesmo resultado que `ft_sort_string_tab()`.

# Capítulo XII

## Submissão e avaliação

Entrega o teu trabalho no teu repositório `Git`, como habitual. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em confirmar os nomes dos teus ficheiros para ter a certeza que estão corretos.



Apenas precisas de entregar os ficheiros pedidos no enunciado deste projeto.