



Piscina C

C 05

Sumário: Este documento é o enunciado do módulo C 05 da Piscina C da 42.

Versão: 7

Conteúdo

I	Instruções	2
II	Intruções IA	4
III	Preâmbulo	7
IV	Exercício 00 : ft_iterative_factorial	9
V	Exercício 01 : ft_recursive_factorial	10
VI	Exercício 02 : ft_iterative_power	11
VII	Exercício 03 : ft_recursive_power	12
VIII	Exercício 04 : ft_fibonacci	13
IX	Exercício 05 : ft_sqrt	14
X	Exercício 06 : ft_is_prime	15
XI	Exercício 07 : ft_find_next_prime	16
XII	Exercício 08 : As dez damas	17
XIII	Submissão e Avaliação	18

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 vai ser lançada com a sinalização `-R CheckForbiddenSourceHeader`. A Moulinette também a utilizará.

Capítulo II

Intruções 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

A seguir, um diálogo tirado do primeiro livro da saga Harry Potter:

Ah, vocês podem me achar pouco atraente,
Mas não me julguem só pela aparência
Engulo a mim mesmo se puderem encontrar
Um chapéu mais inteligente do que o papai aqui.

Podem guardar seus chapéus-coco bem pretos,
Suas cartolas altas de cetim brilhoso
Porque eu sou o Chapéu Seletor de Hogwarts
E dou de dez a zero em qualquer outro chapéu.

Não há nada escondido em sua cabeça
Que o Chapéu Seletor não consiga ver,
Por isso é só me porem na cabeça que vou dizer
Em que casa de Hogwarts deverão ficar.

Quem sabe sua morada é a Grifinória,
Casa onde habitam os corações indomáveis,
Ousadia e sangue-frio e nobreza
Destacam os alunos da Grifinória dos demais.

Quem sabe é na Lufa-Lufa que você vai morar,
Onde seus moradores são justos e leais
Pacientes, sinceros, sem medo da dor;
Ou será a velha e sábia Corvinal,

A casa dos que têm a mente sempre alerta,
Onde os homens de grande espírito e saber
Sempre encontrarão companheiros seus iguais;
Ou quem sabe a Sonserina será a sua casa


E ali fará seus verdadeiros amigos,
Homens de astúcia que usam quaisquer meios
Para atingir os fins que antes colimaram.
Vamos, me experimentem!

Não devem temer! Nem se atrapalhar!
Estarão em boas mãos!
(Mesmo que os chapéus não tenham pés nem mãos)
Porque eu sou único, sou um Chapéu Pensador!

Infelizmente, esse tema não tem nada a ver com a série Harry Potter e é uma pena, porque sua entrega não será feita por magia.

Capítulo IV

Exercício 00 : ft_iterative_factorial


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

- Escreva uma função iterativa que retorne um número. Esse número deve ser o resultado da operação fatorial a partir do número passado como parâmetro.
- Se o argumento não for válido, a função deve retornar 0.
- Não deve gerir o "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo V

Exercício 01 : ft_recursive_factorial


	Exercício : 01
	ft_recursive_factorial
	Pasta de entrega : <i>ex01/</i>
	Ficheiros para entregar : ft_recursive_factorial.c
	Funções autorizadas : Nenhuma

- Escreva uma função recursiva que retorne o fatorial do número passado como parâmetro.
- Se o argumento não for válido, a função deve retornar 0.
- Não deve gerir o "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo VI

Exercício 02 : ft_iterative_power


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

- Escreva uma função iterativa que retorne uma potência de um número. Uma potência inferior a 0 retornará 0.
- Não deve gerir o "int overflow", o retorno da função será indefinido.
- Nós decidimos que a potência de 0 sobre 0 deve retornar 1.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo VII

Exercício 03 : ft_recursive_power


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

- Escreva uma função recursiva que retorne uma potência de um número.
- Uma potência menor que 0 retorna 0.
- Não deve gerir o "int overflow", o retorno da função será indefinido.
- Nós decidimos que a potência de 0 sobre 0 deve retornar 1.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo VIII

Exercício 04 : ft_fibonacci

	Exercício : 04
	ft_fibonacci
	Pasta de entrega : <i>ex04/</i>
	Ficheiros para entregar : <i>ft_fibonacci.c</i>
	Funções autorizadas : Nenhuma


- Escreva uma função `ft_fibonacci` que retorne o *n*-iésimo elemento da sequência de Fibonacci, sendo o primeiro elemento com índice 0. Vamos considerar que a sequência de Fibonacci começa por 0, 1, 1, 2.
- Não deve gerir o "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

- Evidentemente, `ft_fibonacci` deverá ser recursiva.
- Se `index` for inferior a 0, a função retornará -1.

Capítulo IX

Exercício 05 : ft_sqrt


	Exercício : 05
ft_sqrt	
Pasta de entrega : <i>ex05/</i>	
Ficheiros para entregar : ft_sqrt.c	
Funções autorizadas : Nenhuma	

- Escreva uma função que retorne a raiz quadrada inteira de um número se ela existir, e 0 se a raiz quadrada for um número irracional.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo X

Exercício 06 : ft_is_prime

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

- Escreva uma função que retorne 1 se o número recebido for primo e 0 se não for.
- Ela deverá ser prototipada da seguinte maneira:


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



0 e 1 não são números primos.

Capítulo XI

Exercício 07 : ft_find_next_prime


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

- Escreva uma função que retorne o número primo imediatamente superior ou igual ao número passado como parâmetro.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo XII

Exercício 08 : As dez damas

	Exercício : 08
As dez damas	
Pasta de entrega : <i>ex08/</i>	
Ficheiros para entregar : <i>ft_ten_queens_puzzle.c</i>	
Funções autorizadas : <i>write</i>	

- Escreva uma função que mostre todas as possibilidades de posicionar dez damas num tabuleiro de 10x10 sem que elas possam ser comidas com um único movimento e retorne o número de possibilidades.
- É necessário utilizar a recursividade para resolver este exercício.
- Ela deverá ser prototipada da seguinte maneira:

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

- A solução deverá ser imprimida da seguinte maneira:

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

- A sequência é lida da esquerda para a direita. O primeiro número corresponde à posição da primeira Dama na primeira coluna (o índice começando no 0). O enésimo número corresponde à posição da enésima dama na enésima coluna.
- A função tem que retornar o número total de soluções imprimidas.

Capítulo XIII

Submissão e Avaliação

Entrega a tua tarefa no teu repositório `Git` como habitualmente. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em verificar duas vezes os nomes dos teus ficheiros para garantir que estão corretos.



Deves devolver apenas os ficheiros solicitados pelo enunciado deste projeto.