



Piscina C

C 05

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

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : ft_iterative_factorial	6
IV	Exercício 01 : ft_recursive_factorial	7
V	Exercício 02 : ft_iterative_power	8
VI	Exercício 03 : ft_recursive_power	9
VII	Exercício 04 : ft_fibonacci	10
VIII	Exercício 05 : ft_sqrt	11
IX	Exercício 06 : ft_is_prime	12
X	Exercício 07 : ft_find_next_prime	13
XI	Exercício 08 : As dez damas	14

Capítulo I

Instruções

- Somente esta página servirá de referência, não confie nos boatos.
- Leia bem o tema antes de entregar seus exercícios. A qualquer momento o tema pode mudar.
- Atenção aos direitos de seus arquivos e suas pastas.
- Você deve seguir o procedimento de entrega para todos os seus exercícios.
- Os seus exercícios serão corrigidos por seus colegas de piscina.
- Além dos seus colegas, haverá a correção por um programa chamado Moulinette.
- A Moulinette é muito rigorosa na sua avaliação. Ela é completamente automatizada. É impossível discutir sua nota com ela. Tenha um rigor exemplar para evitar surpresas.
- A Moulinette não tem a mente muito aberta. Ela não tenta entender o código que não respeita a Norma. A Moulinette utiliza o programa **norminette** para verificar a norma dos seus arquivos. Então é uma tolice entregar um código que não passa pela **norminette**.
- Os exercícios estão rigorosamente ordenados do mais simples ao mais complexo. Em nenhum caso daremos atenção, nem levaremos em conta um exercício complexo se outro mais simples não tiver sido perfeitamente realizado.
- A utilização de uma função proibida é um caso de fraude. Qualquer fraude é punida com nota de -42.
- Você não deve entregar uma função `main()` se nós pedirmos um programa.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza `gcc`.
- Se o seu programa não compila, você terá 0.

- Você não deve deixar em sua pasta nenhum outro arquivo além daqueles explicitamente especificados pelos enunciados dos exercícios.
- Você tem alguma dúvida? Pergunte ao seu vizinho da direita. Ou tente também perguntar ao seu vizinho da esquerda.
- Seu manual de referência se chama `Google / man / Internet /`
- Considere discutir no fórum Piscina do seu Intra, assim como no slack da sua Piscina!
- Leia atentamente os exemplos. Eles podem muito bem pedir coisas que não estão especificadas no tema...
- Reflita. Por favor, por Odin! Por tudo que é mais sagrado.



Hoje, a Norminette deve ser lançada com a sinalização `-R CheckForbiddenSourceHeader`. A Moulinette também a utilizará.

Capítulo II

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 III

Exercício 00 : ft_iterative_factorial


	Exercício : 00
	ft_iterative_factorial
	Pasta de entrega : ex00/
	Arquivos 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 se deve gerar "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo IV

Exercício 01 : ft_recursive_factorial


	Exercício : 01
	ft_recursive_factorial
	Pasta de entrega : <i>ex01/</i>
	Arquivos 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 se deve gerar "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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


Capítulo V

Exercício 02 : ft_iterative_power


	Exercício : 02
	ft_iterative_power
	Pasta de entrega : <i>ex02/</i>
	Arquivos 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 se deve gerar "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo VI

Exercício 03 : ft_recursive_power


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

- Escreva uma função recursiva que retorne uma potência de um número.
- Não se deve gerar "int overflow", o retorno da função será indefinido.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo VII

Exercício 04 : ft_fibonacci

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


- Escreva uma função `ft_fibonacci` que retorne o *n*-ésimo 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.
- Overflows não devem ser gerados.
- 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 VIII

Exercício 05 : ft_sqrt


	Exercício : 05
	ft_sqrt
	Pasta de entrega : <i>ex05/</i>
	Arquivos 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 não for inteira.
- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo IX

Exercício 06 : ft_is_prime

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

- Escreva uma função que retorne 1 se o número 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 X

Exercício 07 : ft_find_next_prime


	Exercício : 07
	ft_find_next_prime
	Pasta de entrega : <i>ex07/</i>
	Arquivos 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 XI

Exercício 08 : As dez damas

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

- Escreva uma função que mostre todas as possibilidades de posicionar dez damas em um tabuleiro de 10x10 sem que elas possam ser comidas com um único movimento.
- A recursividade deverá ser utilizada.
- O valor de retorno da sua função deverá ser o número de soluções mostradas.
- Ela deverá ser prototipada da seguinte maneira:

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

- A exibição deverá ser feita 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 com 0). O enésimo número corresponde à posição da enésima dama na enésima coluna.