



Piscina C

C 10

Sumário: Este documento é o enunciado do módulo C 10 da Piscina C da 42.

Versão: 5.2

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : display_file	5
IV	Exercício 01 : cat	6
V	Exercício 02 : tail	7
VI	Exercício 03 : hexdump	8
VII	Submissão e avaliação	9

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...

Capítulo II

Preâmbulo


Para começar bem seu dia, veja a seguir algumas questões muito simples:

- O que aconteceria se eu deixasse um secador de cabelo ligado com eletricidade contínua em um cubo hermético de um metro de lado?
- Jogar antimatéria no reator de Chernobyl quando ele estava derretendo teria impedido sua fusão?
- É possível chorar até desidratar?
- Se todos os seres humanos desaparecessem da face da Terra, depois de quanto tempo a última fonte de luz artificial se apagaria?
- É realmente perigoso nadar em uma piscina durante uma tempestade?
- De que altura seria preciso soltar um bife para que ele esteja cozido ao chegar no chão?
- Quando - se é que um dia isso vai acontecer - a largura de banda da Internet vai superar a da FedEx?
- Quantos tweets diferentes são possíveis na nossa língua?
E quanto tempo a população mundial levaria para ler todos em voz alta?
- Qual seria o resultado se todos os candidatos ao exame para tirar carteira de motor respondessem aleatoriamente ao questionário de múltipla escolha?
Quanto acertariam todas as perguntas?

Questões extraídas do livro 'E se?' de Randall Munroe.

Capítulo III

Exercício 00 : display_file

	Exercício : 00
display_file	
Pasta de entrega : <i>ex00/</i>	
Ficheiros para entregar : Makefile, e os arquivos do seu programa	
Funções autorizadas : close, open, read, write	

- Escreva um programa chamado `ft_display_file` que exiba no *standard output* apenas o conteúdo do ficheiro passado como argumento.
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`. O binário chamar-se-á `ft_display_file`.
- A função `malloc` é proibida. Pode fazer o exercício declarando apenas uma matriz de tamanho fixo.
- Todos os ficheiros passados como parâmetro serão válidos.
- As mensagens de erro deverão ser exibidas na saída que lhes está reservada e sendo seguidas por uma quebra de linha.

- Se não houver nenhum argumento, o seu programa deverá mostrar

```
File name missing.
```

- Se existir mais do que um argumento, o seu programa deverá mostrar


```
Too many arguments.
```

- Se o ficheiro não for legível, o seu programa deverá mostrar

```
Cannot read file.
```

Capítulo IV


Exercício 01 : cat

	Exercício : 01
cat	
Pasta de entrega : <code>ex01/</code>	
Ficheiros para entregar : <code>Makefile</code> , e os ficheiros do seu programa	
Funções autorizadas : <code>close</code> , <code>open</code> , <code>read</code> , <code>write</code> , <code>strerror</code> , <code>basename</code>	

- Escreva um programa chamado `ft_cat` que faça o mesmo que o comando `cat` do sistema.
- Não terá que lidar com as opções.
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`.
- Pode utilizar a variável `errno` (ver o `man` de `Errno`).
- Deve ler os `man` de todas as funções autorizadas
- Pode fazer o exercício declarando apenas uma matriz de tamanho fixo. Essa matriz terá um tamanho limitado a um pouco menos de aproximadamente 30 ko. Para testar essa limitação, utilize o comando `ulimit` no seu shell.

Capítulo V


Exercício 02 : tail

	Exercício : 02
tail	
Pasta de entrega : <i>ex02/</i>	
Ficheiros para entregar : Makefile, e os ficheiros do seu programa	
Funções autorizadas : close, open, read, write, malloc, free, strerror, basename	

- Escreva um programa chamado `ft_tail` que execute o mesmo trabalho que o comando `tail`.
- Terá que lidar somente com a opção `-c`, mas não os sinais '+' e '-'.
- Todos os testes serão realizados com a opção `-c`
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`.
- Pode utilizar a variável `errno`

Capítulo VI

Exercício 03 : hexdump

	Exercício : 03
hexdump	
Pasta de entrega : <i>ex03/</i>	
Ficheiros para entregar : <code>Makefile</code> , e os arquivos do seu programa	
Funções autorizadas : <code>close</code> , <code>open</code> , <code>read</code> , <code>write</code> , <code>malloc</code> , <code>free</code> , <code>strerror</code> , <code>basename</code>	

- Escreva um programa chamado `ft_hexdump` que faça o mesmo que o comando `hexdump` do sistema, sem redirecionamento.
- Terá que lidar apenas com a opção `-C`.
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`.
- Pode utilizar a variável `errno` (ver o man de `errno`).

Capítulo VII

Submissão e avaliação

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



É apenas necessário entregar os ficheiros pedidos para este projeto