



Piscina C

C 10

Sumário: Este documento é o tema 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	Entrega e avaliação entre pares	9

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 de 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. Entenda que é uma tolice entregar um código que não passe na **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 `cc`.
- 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 na intranet, assim como no Discord 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.

Capítulo II

Preâmbulo

Body Count é uma banda de heavy metal americana formada em Los Angeles, California, em 1990. O grupo é liderado por Ice-T, que co-fundou o grupo com o guitarrista principal Ernie C pelo interesse de ambos na música heavy metal.

Ice-T tomou o papel de vocalista e compositor da maior parte das letras das canções do grupo. O guitarrista Ernie C foi responsável por compor as músicas do grupo. Seu controverso auto-entitulado álbum de estreia foi lançado pela Sire Records em 1992.


A música "Cop Killer" foi objeto de muita controvérsia. Embora a empresa controladora da Sire Records, a Warner Bros. Records, defendesse o single, Ice-T optou por remover a faixa do álbum porque sentiu que a controvérsia havia eclipsado a própria música. O grupo deixou a Sire no ano seguinte. Desde então, eles lançaram mais três álbuns em gravadoras diferentes, nenhum dos quais foi recebido tão comercialmente ou criticamente quanto seu álbum de estreia.

Três dos seis membros originais da banda já faleceram:
D-Roc morreu de linfoma, Beatmaster V de leucemia e Mooseman num tiroteio.

[Clique aqui](#), comece, e trabalhe... Agora mesmo!

Capítulo III

Exercício 00 : display_file

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

- Escreva um programa chamado `ft_display_file` que exiba na saída padrão somente o conteúdo do arquivo passado como argumento.
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`. O binário vai se chamar `ft_display_file`.
- A função `malloc` é proibida. Você pode fazer o exercício somente declarando uma matriz de tamanho fixo.
- Todos os arquivos 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, seu programa deverá mostrar

```
File name missing.
```

- Se houver muitos argumentos, seu programa devesse mostrar


```
Too many arguments.
```

- Se o arquivo não for legível, seu programa devesse 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 arquivos necessários para o 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 execute o mesmo trabalho que o comando `cat` do sistema.
- Você 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`.
- Você pode utilizar a variável `errno` (ver o `man` de `Errno`).
- Você deve ler os `man` de todas as funções autorizadas
- Você pode fazer o exercício somente declarando 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 arquivos necessários para o 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`.
- Você terá que lidar somente com a opção `-c`, e não o `'+'` e o `'-'`.
- 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`.
- Você pode utilizar a variável `errno`

Capítulo VI

Exercício 03 : hexdump

	Exercício : 03
hexdump	
Pasta de entrega : <code>ex03/</code>	
Ficheiros para entregar : <code>Makefile</code> , e os arquivos necessários para o 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 execute o mesmo trabalho que o comando `hexdump` do sistema sem redirecionamento.
- Você terá que lidar somente com a opção `-C`.
- A pasta de entrega deverá ter um `Makefile` com uma regra `all`, uma regra `clean` e uma regra `fclean`.
- Você pode utilizar a variável `errno` (ver o `man` de `errno`).

Capítulo VII

Entrega e avaliação entre pares

Entregue seu projeto em seu repositório `Git` como de costume. Somente o trabalho contido em seu repositório será avaliado durante a defesa. Não hesite em verificar mais de uma vez os nomes dos seus arquivos para ter certeza de que eles estão corretos.



Você deve submeter somente os arquivos solicitados pelo subject deste projeto.