



Piscina C

C 10

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

Versão: 7.0

Conteúdo

I	Instruções	2
II	Intruções IA	4
III	Foreword	7
IV	Exercício 00 : display_file	8
V	Exercício 01 : cat	9
VI	Exercício 02 : tail	10
VII	Exercício 03 : hexdump	11
VIII	Submissão e avaliação	12

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.
- 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...
- 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

Foreword

Body Count is an American heavy metal band formed in Los Angeles, California, in 1990. The group is fronted by Ice-T, who co-founded the group with lead guitarist Ernie C out of their interest in heavy metal music. Ice-T took on the role of vocalist and writing the lyrics for most of the group's songs. Lead guitarist Ernie C has been responsible for writing the group's music. Their controversial self-titled debut album was released on Sire Records in 1992.


The song "Cop Killer" was the subject of much controversy. Although Sire Records' parent company, Warner Bros. Records, defended the single, Ice-T chose to remove the track from the album because he felt that the controversy had eclipsed the music itself. The group left Sire the following year. Since then, they have released three further albums on different labels, none of which have been received as commercially or critically well as their debut album.

Three out of the band's original six members are deceased: D-Roc died from lymphoma, Beatmaster V from leukemia and Mooseman in a drive-by shooting.

[Click here](#), start it, and work... Right now!

Capítulo IV

Exercício 00 : display_file

	Exercício : 00
display_file	
Pasta de entrega : ex00/	
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 as seguintes regras: `all`, `clean`, `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 V


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.
- O directório de entrega deverá ter um `Makefile` com as seguintes regras: `all`, `clean`, `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 VI


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 VII

Exercício 03 : hexdump

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

- 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 VIII

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