



Piscina C

C 08

Sumário: Este documento é o enuncionado do módulo C 08 da Piscina C da 42.

Versão: 14

Conteúdo

I	Instruções	2
II	Intruções IA	4
III	Preâmbulo	7
IV	Exercício 00 : ft.h	8
V	Exercício 01 : ft_boolean.h	9
VI	Exercício 02 : ft_abs.h	11
VII	Exercício 03 : ft_point.h	12
VIII	Exercício 04 : ft_strs_to_tab	13
IX	Exercício 05 : ft_show_tab	15
X	Submissão e Avaliação	16

Capítulo I

Instruções

- Somente este documento servirá de referência; não confie nos boatos.
- Releia 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

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

Preâmbulo

É isto que a enciclopédia colaborativa *Wikipédia* tem a dizer sobre o ornitorrinco:

O ornitorrinco (*Ornithorhynchus anatinus*) é uma espécie de pequeno mamífero semiaquático endêmico do leste da Austrália, incluindo a Tasmânia. É uma das cinco espécies da ordem dos monotremados, única ordem de mamíferos que põem ovos em vez de dar à luz filhotes completamente formados (as quatro outras espécies são equidnas). É a única espécie sobrevivente da família *Ornithorhynchidae* e do gênero *Ornithorhynchus*, embora um grande número de fragmentos de espécies fósseis dessa família e desse gênero tenha sido descoberto.


A aparência estranha desse mamífero, que põe ovos, tem esporões venenosos, uma mandíbula achatada como um bico de pato, uma cauda que lembra a de um castor, que serve ao mesmo tempo como leme na água e de reserva de gordura, e patas de lontra, realmente surpreendeu os primeiros exploradores que o encontraram. Muitos naturalistas europeus acreditaram que se tratava de uma brincadeira. É um dos raros mamíferos venenosos: o macho tem nas patas traseiras um esporão que pode liberar um veneno capaz de provocar fortes dores a um ser humano. As características incomuns do ornitorrinco o tornaram um objeto importante de estudo para compreender melhor a evolução das espécies animais e o tornaram também um dos símbolos da Austrália: ele foi utilizado como mascote de vários eventos nacionais e está no verso da moeda de 20 centavos australianos.

Até o início do século XX, ele foi caçado por sua pele, mas atualmente é uma espécie protegida. Apesar de os programas de reprodução em cativeiro terem um sucesso muito limitado e o animal ser sensível aos efeitos da poluição, não se considera que a espécie esteja em perigo.

Este subject não tem a ver com os ornitorrincos.

Capítulo IV

Exercício 00 : ft.h


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

- Crie um ficheiro `ft.h`
- Ele deve conter todos os protótipos das funções seguintes:

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

Capítulo V

Exercício 01 : ft_boolean.h

	Exercício : 01
	ft_boolean.h
	Pasta de entrega : ex01/
	Ficheiros para entregar : ft_boolean.h
	Funções autorizadas : Nenhuma

- Escreva um ficheiro `ft_boolean.h` que compile e faça funcionar corretamente o seguinte main:

```
#include "ft_boolean.h"

void      ft_putstr(char *str)
{
    while (*str)
        write(1, str++, 1);
}

t_bool    ft_is_even(int nbr)
{
    return ((EVEN(nbr)) ? TRUE : FALSE);
}

int       main(int argc, char **argv)
{
    (void)argv;
    if (ft_is_even(argc - 1) == TRUE)
        ft_putstr(EVEN_MSG);
    else
        ft_putstr(ODD_MSG);
    return (SUCCESS);
}
```

- Esse programa deverá imprimir

```
I have an even number of arguments.
```

- ou

```
I have an odd number of arguments.
```


- seguido de uma quebra de linha, quando necessário.



A Norminette deve ser lançada com a sinalização `-R CheckDefine`. A Moulinette também a utilizará.

Capítulo VI

Exercício 02 : ft_abs.h

	Exercício : 02
	ft_abs.h
	Pasta de entrega : <i>ex02/</i>
	Ficheiros para entregar : ft_abs.h
	Funções autorizadas : Nenhuma

- Crie uma macro ABS que substitua o parâmetro pelo seu valor absoluto:


```
#define ABS(Value)
```



A Norminette deve ser lançada com a sinalização -R CheckDefine. A Moulinette também a utilizará.

Capítulo VII

Exercício 03 : ft_point.h

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

- Crie um ficheiro `ft_point.h` que compile a seguinte main:

```
#include "ft_point.h"


void      set_point(t_point *point)
{
    point->x = 42;
    point->y = 21;
}

int       main(void)
{
    t_point      point;

    set_point(&point);
    return (0);
}
```

Capítulo VIII

Exercício 04 : ft_strs_to_tab

	Exercício : 04
	ft_strs_to_tab
	Pasta de entrega : ex04/
	Ficheiros para entregar : ft_strs_to_tab.c
	Funções autorizadas : malloc, free

- Escreva uma função que tenha como parâmetros, uma matriz de *string* e o tamanho dessa matriz.
- Ela deverá ser prototipada da seguinte maneira:

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- Ela deve transformar cada elemento da matriz de *string* numa estrutura.
- A estrutura será definida no ficheiro `ft_stock_str.h` da seguinte forma:


```
typedef struct s_stock_str
{
    int size;
    char *str;
    char *copy;
} t_stock_str;
```

- `size` sendo o tamanho da *string*;
 - `str` sendo a *string*;
 - `copy` sendo uma cópia da *string*;
- Ela deve manter a ordem dos elementos de `av`.

- A matriz de estruturas deverá ser alocada, e o último elemento terá 0 como valor de `str` a fim de indicar o fim da matriz.
- Se houver um erro de alocação, ela deverá retornar um *pointer* NULL.
- Vamos testar sua função com o nosso `ft_show_tab` (próximo exercício). Tome as medidas necessárias para que funcione!

Capítulo IX

Exercício 05 : ft_show_tab

	Exercício : 05
	ft_show_tab
	Pasta de entrega : ex05/
	Ficheiros para entregar : ft_show_tab.c
	Funções autorizadas : write

- Escreva uma função que mostre o conteúdo de uma matriz criada pela função anterior.
- Ela deverá ser prototipada da seguinte maneira:

```
void ft_show_tab(struct s_stock_str *par);
```

- A estrutura é a mesma do exercício anterior e estará no ficheiro `ft_stock_str.h` que vamos fornecer:
- Para cada elemento da matriz:
 - a *string* deve ser seguida de uma quebra de linha
 - o tamanho seguido de uma quebra de linha
 - a cópia da *string* (que poderá ter sido modificada) seguida de uma quebra de linha
- Vamos testar sua função com nosso `ft_strs_to_tab` (exercício anterior). Tome as medidas necessárias para que funcione!

Capítulo X

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.