



Piscina C

Sumário: Este documento é o enunciado do módulo C 13 da Piscina C da 42.

Versão: 6.0

Conteúdo

I	Instruções	2
II	Intruções IA	4
III	Preâmbulo	7
IV	Exercício 00 : btree_create_node	8
V	Exercício 01 : btree_apply_prefix	9
VI	Exercise 02 : btree_apply_infix	10
VII	Exercício 03 : btree_apply_suffix	11
VIII	Exercício 04 : btree_insert_data	12
IX	Exercício 05 : btree_search_item	13
X	Exercício 06 : btree_level_count	14
XI	Exercício 07 : btree_apply_by_level	15
XII	Submissão e avaliação	16

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...
- Para os seguintes exercícios, será usada a seguinte estrutura :

```
typedef struct      s_btree
{
    struct s_btree *left;
    struct s_btree *right;
    void *item;
    t_btree;
```

- Deverá incluir esta estrutura no ficheiro `ft_btree.h` e submeter o ficheiro em cada exercício.
- A partir do exercício 01, a função `btree_create_node` será usada regularmente, por isso ajuste o seu código (poderá ser útil ter o seu protótipo no ficheiro `ft_btree.h...`).

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.

During this phase, your main objective is to build your foundation — through struggle, repetition, and especially **peer-learning** exchange.

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


Here's the list of releases for Venom :

- In League with Satan (single, 1980)
- Welcome to Hell (1981)
- Black Metal (1982)
- Bloodlust (single, 1983)
- Die Hard (single, 1983)
- Warhead (single, 1984)
- At War with Satan (1984)
- Hell at Hammersmith (EP, 1985)
- American Assault (EP, 1985)
- Canadian Assault (EP, 1985)
- French Assault (EP, 1985)
- Japanese Assault (EP, 1985)
- Scandinavian Assault (EP, 1985)
- Manitou (single, 1985)
- Nightmare (single, 1985)
- Possessed (1985)
- German Assault (EP, 1987)
- Calm Before the Storm (1987)
- Prime Evil (1989)
- Tear Your Soul Apart (EP, 1990)
- Temples of Ice (1991)
- The Waste Lands (1992)
- Venom '96 (EP, 1996)
- Cast in Stone (1997)
- Resurrection (2000)
- Anti Christ (single, 2006)
- Metal Black (2006)
- Hell (2008)
- Fallen Angels (2011)

Today's subject will seem easier if you listen to Venom.

Capítulo IV

Exercício 00 : btree_create_node


	Exercício : 00
btree_create_node	
Pasta de entrega : <i>ex00/</i>	
Ficheiros para entregar : <code>btree_create_node.c</code> , <code>ft_btree.h</code>	
Funções autorizadas : <code>malloc</code>	

- Escreva a função `btree_create_node` que aloca um novo elemento. Inicialize o novo `item` com o valor do parâmetro passado e todos os outros elementos com 0.
- O endereço do nó criado é retornado.
- Deverá ser prototipada da seguinte forma:

```
t_btree *btree_create_node(void *item);
```

Capítulo V

Exercício 01 : btree_apply_prefix


	Exercício : 01
btree_apply_prefix	
Pasta de entrega : <i>ex01/</i>	
Ficheiros para entregar : <code>btree_apply_prefix.c</code> , <code>ft_btree.h</code>	
Funções autorizadas : Nenhuma	

- Escreva a função `btree_apply_prefix` que aplica a função passada como parâmetro ao item de cada nó, usando `prefix traversal` para percorrer a árvore.
- Deverá ser prototipada da seguinte forma:

```
void btree_apply_prefix(t_btree *root, void (*applyf)(void *));
```

Capítulo VI

Exercise 02 : btree_apply_infix


	Exercício : 02
btree_apply_infix	
Pasta de entrega : <i>ex02/</i>	
Ficheiros para entregar : <code>btree_apply_infix.c</code> , <code>ft_btree.h</code>	
Funções autorizadas : Nenhuma	

- Create a function `btree_apply_infix` which applies the function given as an argument to the `item` of each node, using `infix traversal` to traverse the tree.
- Here's how it should be prototyped :

```
void btree_apply_infix(t_btree *root, void (*applyf)(void *));
```

Capítulo VII

Exercício 03 : btree_apply_suffix


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

- Escreva a função `btree_apply_suffix` que aplica a função passada como parâmetro ao `item` de cada nó, usando `sufix traversal` para percorrer a árvore.
- Deverá ser prototipada da seguinte forma:

```
void btree_apply_suffix(t_btree *root, void (*applyf)(void *));
```

Capítulo VIII

Exercício 04 : btree_insert_data


	Exercício : 04
	btree_insert_data
	Pasta de entrega : <i>ex04/</i>
	Ficheiros para entregar : btree_insert_data.c , ft_btree.h
	Funções autorizadas : btree_create_node

- Escreva a função **btree_insert_data** que insere o elemento **item** em uma árvore. A árvore passada como parâmetro será organizada: por cada nó todos os elementos inferiores encontram-se à esquerda e todos os elementos superiores ou iguais, à direita. Será enviada como parâmetro uma função de comparação que tem o mesmo comportamento que **strcmp**.
- O parâmetro **root** aponta para o nó raiz da árvore. Na primeira chamada, deverá apontar para **NULL**.
- Deverá ser prototipada da seguinte forma:

```
void btree_insert_data(t_btree **root, void *item, int (*cmpf)(void *, void *));
```

Capítulo IX

Exercício 05 : btree_search_item


	Exercício : 05
	btree_search_item
	Pasta de entrega : <i>ex05/</i>
	Ficheiros para entregar : <code>btree_search_item.c</code> , <code>ft_btree.h</code>
	Funções autorizadas : Nenhuma

- Escreva a função `btree_search_item` que retorna o primeiro elemento correspondente ao dado de referência passado como parâmetro. A árvore deverá ser percorrida usando `infix traversal`. Se o elemento não for encontrado, a função deverá retornar `NULL`.
- Deverá ser prototipada da seguinte forma:

```
void *btree_search_item(t_btree *root, void *data_ref, int (*cmpf)(void *, void *));
```

Capítulo X

Exercício 06 : btree_level_count


	Exercício : 06
	btree_level_count
	Pasta de entrega : <i>ex06/</i>
	Ficheiros para entregar : btree_level_count.c , ft_btree.h
	Funções autorizadas : Nenhuma

- Escreva a função `btree_level_count` que retorna a profundidade do maior ramo passado como parâmetro.
- Deverá ser prototipada da seguinte forma:

```
int btree_level_count(t_btree *root);
```

Capítulo XI

Exercício 07 : btree_apply_by_level

	Exercício : 07
	btree_apply_by_level
	Pasta de entrega : <i>ex07/</i>
	Ficheiros para entregar : <i>btree_apply_by_level.c, ft_btree.h</i>
	Funções autorizadas : <i>malloc, free</i>

- Escreva a função `btree_apply_by_level` que aplica a função passada como parâmetro a cada nó da árvore. A árvore deve percorrer nível de profundidade por nível de profundidade. A função chamada terá três parâmetros:
 - O primeiro parâmetro, do tipo `void *`, corresponde ao item do nó;
 - O segundo parâmetro, do tipo `int`, corresponde ao nível de profundidade no qual estamos: 0 para a raiz, 1 para seus filhos, 2 para seus netos, etc. ;
 - O terceiro parâmetro, do tipo `int`, vale 1 se for o primeiro nó do nível, caso contrário vale 0.
- Deverá ser prototipada da seguinte forma:

```
void btree_apply_by_level(t_btree *root, void (*applyf)(void *item, int current_level, int is_first_elem))
```


Capítulo XII

Submissão e avaliação

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



Apenas precisas de entregar os ficheiros pedidos no enunciado deste projeto.