



Piscina C

C 00

Sumário: ESTE documento é o enunciado do módulo C 00 da Piscina C da 42.

Versão: 7.7

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00: ft_putchar	5
IV	Exercício 01: ft_print_alphabet	6
V	Exercício 02: ft_print_reverse_alphabet	7
VI	Exercício 03: ft_print_numbers	8
VII	Exercício 04: ft_is_negative	9
VIII	Exercício 05: ft_print_comb	11
IX	Exercício 06: ft_print_comb2	12
X	Exercício 07: ft_putnbr	13
XI	Exercício 08: ft_print_combn	14
XII	Submissão e avaliação	15

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...



Não se esqueça de adicionar o *standard 42 header* nos seus ficheiros `.c/.h`. A *norminette* verifica a sua existência de qualquer forma!



A *Norminette* deve ser lançada com a *flag* `-R CheckForbiddenSourceHeader`. A *Moulinette* também a utilizará.

Capítulo II

Preâmbulo

Cod liver oil is a nutritional supplement derived from liver of cod fish (Gadidae).

As with most fish oils, it has high levels of the omega-3 fatty acids, eicosapentaenoic acid (EPA) and docosahexaenoic acid (DHA). Cod liver oil also contains vitamin A and vitamin D.


It has historically been taken because of its vitamin A and vitamin D content.

It was once commonly given to children, because vitamin D has been shown to prevent rickets and other symptoms of vitamin D deficiency.

Contrary to Cod liver oil, C is good, eat some!

Capítulo III

Exercício 00: ft_putchar

	Exercício : 00
	ft_putchar
	Pasta de entrega : <i>ex00/</i>
	Ficheiros para entregar : ft_putchar.c
	Funções autorizadas : write

- Escreva uma função que mostre o caractere passado como parâmetro.
- Deve ser prototipada da seguinte maneira:

```
void ft_putchar(char c);
```

Para mostrar o caractere, deve usar a função **write** da seguinte maneira.


```
write(1, &c, 1);
```



O prazo da primeira tentativa é curto, não hesite em efetuar uma avaliação intermédia para medir os seus progressos.

Capítulo IV

Exercício 01: ft_print_alphabet

	Exercício : 01
	ft_print_alphabet
	Pasta de entrega : <i>ex01/</i>
	Ficheiros para entregar : <code>ft_print_alphabet.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre o alfabeto em minúsculo, numa única linha, em ordem crescente, começando pela letra 'a'.
- Deve ser prototipada da seguinte maneira:

```
void ft_print_alphabet(void);
```




Não hesite em contactar aleatoriamente alguém no cluster para fazer uma pergunta.

Capítulo V

Exercício 02:

ft_print_reverse_alphabet

	Exercício : 02
	ft_print_reverse_alphabet
	Pasta de entrega : <i>ex02/</i>
	Ficheiros para entregar : <code>ft_print_reverse_alphabet.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre o alfabeto em minúsculo, numa única linha, em ordem decrescente, começando pela letra 'z'.
- Deve ser prototipada da seguinte maneira:


```
void ft_print_reverse_alphabet(void);
```



Git push regularmente.

Capítulo VI

Exercício 03: ft_print_numbers

	Exercício : 03
	ft_print_numbers
	Pasta de entrega : <i>ex03/</i>
	Ficheiros para entregar : <code>ft_print_numbers.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre todos os números, numa única linha, em ordem crescente.
- Deve ser prototipada da seguinte maneira:


```
void ft_print_numbers(void);
```



Colaboração é a chave do sucesso.

Capítulo VII

Exercício 04: ft_is_negative

	Exercício : 04
	ft_is_negative
	Pasta de entrega : <i>ex04/</i>
	Ficheiros para entregar : <code>ft_is_negative.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre 'N' ou 'P' segundo o sinal do inteiro passado como parâmetro. Se `n` for negativo, indique 'N'. Se `n` for positivo ou nulo, indique 'P'.
- Deve ser prototipada da seguinte maneira:

```
void ft_is_negative(int n);
```



Falhar faz parte do processo de aprendizagem.

Milestone Achieved, Keep Going!

Chegou ao fim dos exercícios obrigatórios para validar este projeto.


Cabe-lhe a si decidir se pretende continuar com os seguintes exercícios opcionais ou passar para o seu próximo projeto. Ambos os caminhos farão com que vejas elementos úteis um dia ou outro.

Para fazeres a tua escolha, tem em conta os seguintes elementos:

- O primeiro exame é sobre programação em C. Por isso, é possível que já tenhas experimentado o primeiro projeto em C. O mesmo se aplica à corrida no final da semana (vais aprender em breve sobre a corrida).
- A sua excelência nesta Piscina será avaliada com base em vários factores. A conclusão de cada projeto é um deles, mas o progresso global ao longo de toda a lista de projectos da Piscina é outro. Escolha sabiamente para otimizar os seus resultados.
- Será sempre possível voltar a tentar o mesmo projeto dentro de alguns dias/semanas, até ao final da Piscina.
- Manter-se sincronizado com os seus pares garante uma melhor colaboração.

Capítulo VIII

Exercício 05: ft_print_comb

	Exercício : 05
	ft_print_comb
	Pasta de entrega : <i>ex05/</i>
	Ficheiros para entregar : <code>ft_print_comb.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre, em ordem crescente, todas as diferentes combinações de três números diferentes em ordem crescente - sim, a repetição é intencional.
- O resultado esperado seria:

```
$>./a.out | cat -e
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$>
```

- 987 fica de fora porque 789 já está presente
- 999 fica de fora porque não tem exclusivamente dígitos diferentes uns dos outros.
- Deve ser prototipada da seguinte maneira:


```
void ft_print_comb(void);
```



Consultou o seu vizinho do lado direito ?

Capítulo IX

Exercício 06: ft_print_comb2

	Exercício : 06
	ft_print_comb2
	Pasta de entrega : <i>ex06/</i>
	Ficheiros para entregar : <code>ft_print_comb2.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre todas as diferentes combinações de dois números (XX XX) entre 0 e 99, em ordem crescente.
- O resultado esperado seria:

```
$>./a.out | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$>
```

- Deve ser prototipada da seguinte maneira:


```
void ft_print_comb2(void);
```



Inspire-se nos outros, não deixe que eles façam o seu trabalho.

Capítulo X

Exercício 07: ft_putnbr

	Exercício : 07
	ft_putnbr
	Pasta de entrega : <i>ex07/</i>
	Ficheiros para entregar : ft_putnbr.c
	Funções autorizadas : write

- Escreva uma função que mostre um número, passado como parâmetro. A função deverá ser capaz de representar todos os valores possíveis de uma variável do tipo `int`.
- Deve ser prototipada da seguinte maneira:

```
void ft_putnbr(int nb);
```


- Por exemplo:
 - `ft_putnbr(42)` mostra "42".



Não acredite em qualquer fonte de informação: faça sempre os seus próprios testes, controlos e verificações.

Capítulo XI

Exercício 08: ft_print_combn

	Exercício : 08
	ft_print_combn
	Pasta de entrega : <i>ex08/</i>
	Ficheiros para entregar : <code>ft_print_combn.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que mostre todas as diferentes combinações de `n` números em ordem crescente.
- `n` será tal que: $0 < n < 10$.
- Se `n = 2`, o resultado esperado seria:

```
$>./a.out | cat -e
01, 02, 03, ..., 09, 12, ..., 79, 89$>
```

- Deve ser prototipada da seguinte maneira:

```
void ft_print_combn(int n);
```



Verificaste com o teu vizinho da esquerda ?

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.