

Piscina C C 07

Sumário: Este documento é o enunciado do módulo C 07 da Piscina C da 42.

Versão: 4.3

# Conteúdo

1	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : ft_strdup	Ę
IV	Exercício 01 : ft_range	6
$\mathbf{V}$	Exercício 02 : ft_ultimate_range	7
VI	Exercício 03 : ft_strjoin	8
VII	Exercício 04 : ft_convert_base	ξ
VIII	Exercício 05 : ft_split	10
$\mathbf{IX}$	Submissão e avaliação	

#### 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 textitflags -Wall -Wextra -Werror, e utiliza cc.
- Se o seu programa não compila, terá 0.
- <u>Não deve</u> deixar no repositório de entrega <u>nenhum</u> 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

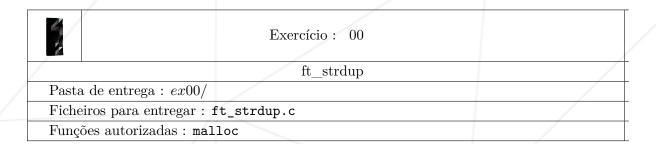
#### Preâmbulo

Veja a seguir uma lista de monstros que podem ser encontrados no famoso Donjon de Naheulbeuk:

- Vários tipos de mortos-vivos;
- Aranhas gigantes;
- Orcs;
- Goblins;
- Trolls no subsolo;
- Feiticeiros;
- Guerreiros amaldiçoados;
- Ratos mutantes;
- Uma garrafa de óleo;
- Papel higiênico;
- Duas esponjas;
- Raviolis.

# Capítulo III

# Exercício 00 : ft\_strdup

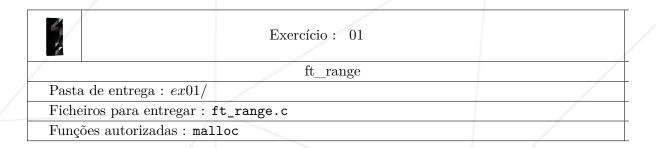


- Reproduzir de forma idêntica o funcionamento da função strdup (man strdup).
- Deverá ser prototipada da seguinte maneira:

char \*ft\_strdup(char \*src);

#### Capítulo IV

#### Exercício 01: ft\_range



- Escreva uma função ft\_range que retorna um *array* de int. Este *array* de int deverá conter todos os valores entre min e max.
- Min incluído max excluído.
- Deverá ser prototipada da seguinte maneira:

int \*ft\_range(int min, int max);

 $\bullet\,$  Se o valor  $\mathtt{min}$  for superior ou igual ao valor  $\mathtt{max},$ um pointernulo será devolvido.

#### Capítulo V

# Exercício 02: ft\_ultimate\_range

	Exercício: 02	
/	ft_ultimate_range	/
Pasta de entrega : $ex02/$		
Ficheiros para entregar : ft_ultimate_range.c		
Funções autorizadas : malloc		

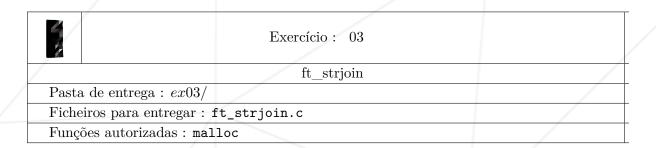
- Escreva uma função ft\_ultimate\_range que aloque e atribua um array de int. Este array de int deverá conter todos os valores entre min e max.
- Min incluído max excluído.

```
int ft_ultimate_range(int **range, int min, int max);
```

- O parâmetro range será devolvido (ou -1 se houver um problema).
- Se o valor min for superior ou igual ao valor max, um *pointer* nulo será devolvido e deve retornar 0.

#### Capítulo VI

# Exercício 03 : ft\_strjoin

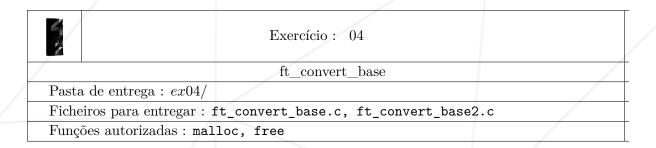


- Escreva uma função que vai concatenar o conjunto das *strings* apontadas por **strs** separando-as com **sep**.
- O parâmetro size é obtido através do número de strings em strs.
- Se size for 0, é preciso devolver uma string elegível de ser argumento para free().
- Deverá ser prototipada da seguinte maneira:

char \*ft\_strjoin(int size, char \*\*strs, char \*sep);

#### Capítulo VII

#### Exercício 04 : ft\_convert\_base

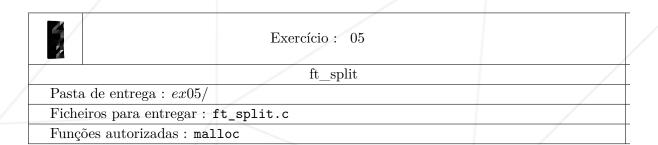


- Escreva uma função que devolva o resultado da conversão da *string* nbr expressa de uma base base\_from noutra base base\_to.
- nbr, base\_from, base\_to não serão necessariamente editáveis.
- nbr seguirá as mesmas regras que ft\_atoi\_base. Por isso, cuidado com '+', '-', e espaços.
- O número representado por nbr cabe num int.
- Se uma base estiver incorreta, a função devolverá NULL.
- O número devolvido deve ser prefixado por apenas um '-' se for necessário; nada de espaços, ou de +.
- Deverá ser prototipada da seguinte maneira:

char \*ft\_convert\_base(char \*nbr, char \*base\_from, char \*base\_to);

#### Capítulo VIII

Exercício 05 : ft\_split



- Escreva uma função que divida uma string em função de uma outra string.
- Será necessário utilizar cada caracter da string charset como separador.
- A função devolve um *array* onde cada seu elemento contém o endereço de uma *string* originalmente compreendida entre dois separadores. O último elemento do *array* deverá ser igual a 0 para marcar o fim do *array*.
- Não deve haver *strings vazias* no seu *array*. Tire as conclusões necessárias.
- A string recebida como parâmetro não será editável.
- Deverá ser prototipada da seguinte maneira:

char \*\*ft\_split(char \*str, char \*charset);

# Capítulo IX

#### 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