



Piscina C

C 07

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

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : ft_strdup	5
IV	Exercício 01 : ft_range	6
V	Exercício 02 : ft_ultimate_range	7
VI	Exercício 03 : ft_strjoin	8
VII	Exercício 04 : ft_convert_base	9
VIII	Exercício 05 : ft_split	10

Capítulo I

Instruções

- Somente esta página servirá de referência, não confie nos boatos.
- Leia bem o tema antes de entregar seus exercícios. A qualquer momento o tema pode mudar.
- Atenção aos direitos de seus arquivos e suas pastas.
- Você deve seguir o procedimento de entrega para todos os seus exercícios.
- Os seus exercícios serão corrigidos por seus colegas de piscina.
- Além dos seus colegas, haverá a correção por um programa chamado Moulinette.
- A Moulinette é muito rigorosa na sua avaliação. Ela é completamente automatizada. É impossível discutir sua nota com ela. Tenha um rigor exemplar para evitar surpresas.
- A Moulinette não tem a mente muito aberta. Ela não tenta entender o código que não respeita a Norma. A Moulinette utiliza o programa **norminette** para verificar a norma dos seus arquivos. Então é uma tolice entregar um código que não passa pela **norminette**.
- Os exercícios estão rigorosamente ordenados do mais simples ao mais complexo. Em nenhum caso daremos atenção, nem levaremos em conta um exercício complexo se outro mais simples não tiver sido perfeitamente realizado.
- A utilização de uma função proibida é um caso de fraude. Qualquer fraude é punida com nota de -42.
- Você não deve entregar uma função `main()` se nós pedirmos um programa.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza `gcc`.
- Se o seu programa não compila, você 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...
- Reflita. Por favor, por Odin! Por tudo que é mais sagrado.



Hoje, a Norminette deve ser lançada com a sinalização `-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


	Exercício : 00
	ft_strdup
	Pasta de entrega : <i>ex00/</i>
	Arquivos para entregar : <code>ft_strdup.c</code>
	Funções autorizadas : <code>malloc</code>

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

```
char *ft_strdup(char *src);
```

Capítulo IV

Exercício 01 : ft_range

	Exercício : 01
	ft_range
	Pasta de entrega : <i>ex01/</i>
	Arquivos para entregar : ft_range.c
	Funções autorizadas : malloc


- Escreva uma função **ft_range** que inverta uma matriz de **int**. Essa matriz de **int** deverá conter todos os valores entre **min** e **max**.
- **Min** incluído - **max** excluído.
- Ela deverá ser prototipada da seguinte maneira:

```
int *ft_range(int min, int max);
```

- Se o valor **min** for superior ou igual ao valor **max**, um ponteiro nulo será retornado.

Capítulo V

Exercício 02 : ft_ultimate_range

	Exercício : 02
	ft_ultimate_range
	Pasta de entrega : <i>ex02/</i>
	Arquivos para entregar : <code>ft_ultimate_range.c</code>
	Funções autorizadas : <code>malloc</code>


- Escreva uma função `ft_ultimate_range` que aloque e atribua uma matriz de `int`. Essa matriz de `int` deverá conter todos os valores entre `min` e `max`.
- Min inclu - max exclu.
- Ela deverá ser prototipada da seguinte maneira:

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

- O tamanho de `range` será retornado (ou -1 se houver um problema).
- Se o valor `min` for superior ou igual ao valor `max`, `range` irá apontar para NULL e retornaremos 0.

Capítulo VI

Exercício 03 : ft_strjoin


	Exercício : 03
	ft_strjoin
	Pasta de entrega : <i>ex03/</i>
	Arquivos para entregar : <code>ft_strjoin.c</code>
	Funções autorizadas : <code>malloc</code>

- Escreva uma função que vai concatenar o conjunto das cadeias de caracteres apontadas por `strs` separando-as com `sep`.
- `size` representado pelo tamanho de `strs`.
- Se `size` valer 0, é preciso retornar uma cadeia de caracteres que podemos `free()`.
- Ela deverá ser prototipada da seguinte maneira:

```
char *ft_strjoin(int size, char **strs, char *sep);
```

Capítulo VII

Exercício 04 : ft_convert_base


	Exercício : 04
	ft_convert_base
	Pasta de entrega : <i>ex04/</i>
	Arquivos para entregar : <code>ft_convert_base.c</code> , <code>ft_convert_base2.c</code>
	Funções autorizadas : <code>malloc</code> , <code>free</code>

- Escreva uma função que retorne o resultado da conversão da cadeia `nbr` expressa em uma base `base_from` em uma 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 whitespaces.
- O número representado por `nbr` cabe em um `int`.
- Se uma base estiver incorreta, a função irá retornar `NULL`.
- O número retornado deve ser prefixado por apenas um '-' se for necessário, nada de whitespaces ou de '+'
- Ela deverá ser prototipada da seguinte maneira:

```
char *ft_convert_base(char *nbr, char *base_from, char *base_to);
```

Capítulo VIII

Exercício 05 : ft_split

	Exercício : 05
	ft_split
	Pasta de entrega : <i>ex05/</i>
	Arquivos para entregar : ft_split.c
	Funções autorizadas : malloc

- Escreva uma função que divida uma cadeia de caracteres em função de uma outra cadeia de caracteres.
- Será necessário utilizar cada caractere da cadeia **charset** como separador.
- A função que retorna uma matriz ou cada elemento desta contém o endereço de uma cadeia de caracteres compreendida entre dois separadores. O último elemento da matriz deverá ser igual a 0 para marcar o fim da matriz.
- Não deve haver cadeias vazias na sua matriz. Tire as conclusões necessárias.
- A cadeia que será transmitida não será editável.
- Ela deverá ser prototipada da seguinte maneira:

```
char **ft_split(char *str, char *charset);
```