



# Piscina C

## C 04

*Sumário: Este documento é o tema do módulo C 04 da Piscina C da 42.*

# Conteúdo

<b>I</b>	<b>Instruções</b>	<b>2</b>
<b>II</b>	<b>Preâmbulo</b>	<b>4</b>
<b>III</b>	<b>Exercício 00 : ft_strlen</b>	<b>5</b>
<b>IV</b>	<b>Exercício 01 : ft_putstr</b>	<b>6</b>
<b>V</b>	<b>Exercício 02 : ft_putnbr</b>	<b>7</b>
<b>VI</b>	<b>Exercício 03 : ft_atoi</b>	<b>8</b>
<b>VII</b>	<b>Exercício 04 : ft_putnbr_base</b>	<b>10</b>
<b>VIII</b>	<b>Exercício 05 : ft_atoi_base</b>	<b>12</b>

# 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 não 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 normete deve ser lançada com a sinalização `-R CheckForbiddenSourceHeader`. A Moulinette também a utilizará.

# Capítulo II

## Preâmbulo

Veja a seguir a letra da música de abertura de Nicky Larson:

Uma sombra corre na noite  
É um assassino fugindo  
E como um demônio ele sorri  
Seu crime ficará impune  
Um carro aparece  
Uma freada brusca, pneus que cantam  
Um tiro que ressoa  
A justiça se chama Nicky

[Refrão]  
No calor  
Da noite  
O mal sempre é punido  
Nenhum perigo o assusta  
Ele gosta de desafios  
E é apaixonado pela justiça  
Nicky Larson não teme ninguém  
Quando os tiros ressoam  
Como um relâmpago ele irrompe  
Principalmente se for uma bela jovem  
Nicky Larson não teme ninguém


Como um caçador ele persegue sua presa  
Para que a justiça e o direito  
Triunfem, ele está prestes a doar  
Toda sua vida sem hesitar  
Quando sua sombra aparece  
Os bandidos começam a tremer  
Eles sabem que nunca vão conseguir  
Escapar dessa justiça

[Refrão]

Infelizmente, esse tema não tem nada a ver com Nicky Larson.

# Capítulo III

## Exercício 00 : ft\_strlen


	Exercício : 00
ft_strlen	
Pasta de entrega : ex00/	
Arquivos para entregar : ft_strlen.c	
Funções autorizadas : Nenhuma	

- Escreva uma função que contenha o número de caracteres em uma cadeia de caracteres e que retorne o número encontrado.
- Ela deverá ser prototipada da seguinte maneira:

```
int    ft_strlen(char *str);
```

# Capítulo IV

## Exercício 01 : ft\_putstr


	Exercício : 01
	ft_putstr
	Pasta de entrega : <i>ex01/</i>
	Arquivos para entregar : <b>ft_putstr.c</b>
	Funções autorizadas : <b>write</b>

- Escreva uma função que mostre um a um os caracteres de uma string na tela.
- O endereço do primeiro caractere da cadeia está contido no ponteiro passado como parâmetro na função.
- Ela deverá ser prototipada da seguinte maneira:

```
void    ft_putstr(char *str);
```

# Capítulo V

## Exercício 02 : ft\_putnbr

	Exercício : 02
	ft_putnbr
	Pasta de entrega : <i>ex02/</i>
	Arquivos para entregar : <b>ft_putnbr.c</b>
	Funções autorizadas : <b>write</b>

- Escreva uma função que mostre um número passado como parâmetro. A função deverá ser capaz de mostrar a totalidade dos valores possíveis em uma variável de tipo `int`.
- Ela deverá ser prototipada da seguinte maneira:


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

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



# Capítulo VI

## Exercício 03 : ft\_atoi

	Exercício : 03
	ft_atoi
	Pasta de entrega : <i>ex03/</i>
	Arquivos para entregar : <i>ft_atoi.c</i>
	Funções autorizadas : Nenhuma

- Escreva uma função que converta o início da cadeia apontada por `str` em inteiro do tipo `int`
- `str` pode começar por um número arbitrário de 'white space' (como definido por `isspace(3)`)
- `str` pode então ser seguido por um número arbitrário de sinais `+` e sinais `-`. O sinal `-` vai mudar o sinal do inteiro retornado em função do número de sinais `-` e se este for par ou ímpar.
- Por fim, `str` deverá ser composta por número da base 10
- Sua função deverá ler `str` desde que esta siga as regras acima e ela deve retornar o número encontrado até então.
- Você não deve considerar os overflows e os underflows, o resultado é considerado indefinido nesses casos.
- Você pode comparar sua função com a verdadeira função `atoi`, exceto a parte dos sinais bem como do overflow.
- Veja um exemplo de um programa que mostra o valor de retorno de `atoi`:


```
$> ./a.out " ---+---+1234ab567"  
-1234
```

- Ela deverá ser prototipada da seguinte maneira:

```
int    ft_atoi(char *str);
```

# Capítulo VII

## Exercício 04 : ft\_putnbr\_base


	Exercício : 04
	ft_putnbr_base
	Pasta de entrega : <i>ex04/</i>
	Arquivos para entregar : <b>ft_putnbr_base.c</b>
	Funções autorizadas : <b>write</b>

- Escreva uma função que mostre um número no terminal em uma base dada.
- Esse número é fornecido como um `int` e a base como uma **cadeia de caracteres**.
- A base contém todos os símbolos utilizáveis para mostrar o número:
  - 0123456789 é a base geralmente utilizada para representar nossos números decimais;
  - 01 é uma base binária;
  - 0123456789ABCDEF é uma base hexadecimal;
  - poneyvif é uma base octal.
- A função deve lidar com os números negativos.
- Se um parâmetro contiver um erro, a função não mostrará nada. Um erro pode ser:
  - a base é vazia ou de tamanho 1;
  - a base contém duas vezes o mesmo caractere;
  - a base contém os caracteres + ou -.
- Ela deverá ser prototipada da seguinte maneira:

```
void      ft_putnbr_base(int nbr, char *base);
```

# Capítulo VIII

## Exercício 05 : ft\_atoi\_base

	Exercício : 05
	ft_atoi_base
	Pasta de entrega : <i>ex05/</i>
	Arquivos para entregar : <code>ft_atoi_base.c</code>
	Funções autorizadas : Nenhuma

- Escreva uma função que converta o início de uma cadeia apontada por `str` em inteiro do tipo `int`.
- `str` está em uma base específica passada como segundo argumento da função.
- Exceto o sistema de base, essa função deve reproduzir o comportamento de `ft_atoi`.
- Se um parâmetro contiver um erro, a função retornará 0. Um erro pode ser:
  - a base é vazia ou de tamanho 1;
  - a base contém duas vezes o mesmo caractere;
  - a base contém os caracteres `+` ou `-` ou whitespaces;
- Ela deverá ser prototipada da seguinte maneira:

```
int      ft_atoi_base(char *str, char *base);
```