



# Piscina C

## C 04

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

*Versão: 4.2*

# 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>6</b>
<b>IV</b>	<b>Exercício 01 : ft_putstr</b>	<b>7</b>
<b>V</b>	<b>Exercício 02 : ft_putnbr</b>	<b>8</b>
<b>VI</b>	<b>Exercício 03 : ft_atoi</b>	<b>9</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>
<b>IX</b>	<b>Submissão e avaliação</b>	<b>13</b>

# Capítulo I

## Instruções

- Somente este documento servirá de referência; não confie nos boatos.
- Leia bem o enunciado antes de entregar seus exercícios. A qualquer momento pode haver alterações.
- 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.
- Tenha atenção aos direitos dos seus ficheiros 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, sejam rigorosos!
- Os exercícios de shell devem ser executados com `/bin/sh`.
- Não deve deixar no repositório de entrega nenhum outros ficheiros além daqueles explicitamente especificados pelos enunciados 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...



Hoje, a normete deve ser lançada com a sinalização -R  
CheckForbiddenSourceHeader. A Moulinette também a utilizará.

# Capítulo II

## Preâmbulo

Here are the lyrics for City Hunter's theme song "Moonlight Shadow":

The last time ever she saw him  
Carried away by a moonlight shadow  
He passed on worried and warning  
Carried away by a moonlight shadow.  
Lost in a riddle that Saturday night  
Far away on the other side.  
He was caught in the middle of a desperate fight  
And she couldn't find how to push through

The trees that whisper in the evening  
Carried away by a moonlight shadow  
Sing a song of sorrow and grieving  
Carried away by a moonlight shadow  
All she saw was a silhouette of a gun  
Far away on the other side.  
He was shot six times by a man on the run  
And she couldn't find how to push through

[Chorus]  
I stay, I pray  
See you in Heaven far away...  
I stay, I pray  
See you in Heaven one day.

Four A.M. in the morning  
Carried away by a moonlight shadow  
I watched your vision forming  
Carried away by a moonlight shadow  
A star was glowing in the silvery night  
Far away on the other side  
Will you come to talk to me this night  
But she couldn't find how to push through


[Chorus]

Far away on the other side.  
Caught in the middle of a hundred and five  
The night was heavy and the air was alive  
But she couldn't find how to push through  
Carried away by a moonlight shadow  
Carried away by a moonlight shadow  
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

# Capítulo III

## Exercício 00 : ft\_strlen


	Exercício : 00
	ft_strlen
	Pasta de entrega : <i>ex00/</i>
	Ficheiros para entregar : <b>ft_strlen.c</b>
	Funções autorizadas : <b>Nenhuma</b>

- Escreve uma função que conte e retorne o numero de caracteres numa string.
- Deve 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>
	Ficheiros para entregar : <b>ft_putstr.c</b>
	Funções autorizadas : <b>write</b>


- Escreve uma função que mostre uma string de caracteres no standard output.
- Deve 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>	
Ficheiros para entregar : <b>ft_putnbr.c</b>	
Funções autorizadas : <b>write</b>	


- Escreve uma função que mostre o número passado como parâmetro. A função deverá ser capaz de mostrar a totalidade dos valores possíveis em uma variável de tipo `int`.
- Deve 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>
	Ficheiros para entregar : <b>ft_atoi.c</b>
	Funções autorizadas : <b>Nenhuma</b>

- Escreve uma função que converta o início da string para a sua representação em inteiro
- A string pode começar por um número arbitrário de 'white space' (como definido por o `isspace(3)`)
- A string pode ser seguido por um número arbitrário de sinais + ou -. O sinal - vai mudar o sinal do inteiro retornado baseado no número de sinais - for par ou ímpar.
- Por fim, a string deverá ser seguido por qualquer número da base 10
- Sua função deve ler a string, desde que a string siga as regras e retorne o número encontrado até então.
- Não debes considerar os overflows ou underflows, o resultado é considerado indefinido nesses casos.
- Veja um exemplo de um programa que mostra o retorno do atoi:


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

- Deve 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>	
Ficheiros para entregar : <b>ft_putnbr_base.c</b>	
Funções autorizadas : <b>write</b>	

- Escreve uma função que mostre o número no terminal em uma base passada como parâmetro.
- Esse número é fornecido como um `int` e a base como uma `string`.
- A base contém todos os símbolos utilizáveis para mostrar o número:
  - 0123456789 é a base geralmente utilizada para representar 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 vai mostrar nada. Um erro pode ser:
  - base é vazia ou de tamanho 1;
  - base contém duas vezes o mesmo caractere;
  - base contém os caracteres + ou -.
- Deve 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>
	Ficheiros para entregar : <b>ft_atoi_base.c</b>
	Funções autorizadas : <b>Nenhuma</b>

- Escreve uma função que converta o início da string para a sua representação em inteiro.
- A string está numa base específica passada como segundo argumento.
- Exceto o sistema de base, essa função deve replicar o comportamento do `ft_atoi`.
- Se um parâmetro contiver um erro, a função retorna 0. Um erro pode ser:
  - base é vazia ou de tamanho 1;
  - base contém duas vezes o mesmo caractere;
  - base contém os caracteres + ou - ou whitespaces;
- Deve ser prototipada da seguinte maneira:

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

# Capítulo IX

## 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.