



Piscina C

C 02

Sumário: Este documento é o tema do módulo C 02 da Piscina C da 42.

Versão: 5.4

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : ft_strcpy	6
IV	Exercício 01 : ft_strncpy	7
V	Exercício 02 : ft_str_is_alpha	8
VI	Exercício 03 : ft_str_is_numeric	9
VII	Exercício 04 : ft_str_is_lowercase	10
VIII	Exercício 05 : ft_str_is_uppercase	11
IX	Exercício 06 : ft_str_is_printable	12
X	Exercício 07 : ft_strupcase	13
XI	Exercício 08 : ft_strlowcase	14
XII	Exercício 09 : ft_strcapitalize	15
XIII	Exercício 10 : ft_strlcpy	16
XIV	Exercício 11 : ft_putstr_non_printable	17
XV	Exercício 12 : ft_print_memory	18
XVI	Entrega e avaliação entre pares	20

Capítulo I

Instruções

- Somente esta página servirá de referência, não confie nos boatos.
- Cuidado! Este documento pode mudar até uma hora antes da entrega.
- Certifique-se que você tenha as permissões apropriadas nos seus arquivos e pastas.
- Você deverá seguir o procedimento de entrega para todos os exercícios.
- Os seus exercícios serão corrigidos por seus colegas de Piscina.
- Além disso, seus exercícios serão verificados e corrigidos por um programa chamado Moulinette
- A Moulinette é extremamente rígida em sua avaliação. Ela é completamente automatizada e é impossível discutir sua nota com ela. Se você quiser evitar surpresas, tenha muita atenção.
- A Moulinette não tem a mente muito aberta. Ela não tentará entender seu código se ele não respeitar a Norma. A Moulinette conta com um programa chamado **norminette** para verificar se seus arquivos respeitam a Norma. TL;DR: seria idiota enviar um trabalho que não passa na verificação da **norminette**.
- Os exercícios estão arranjados cuidadosamente em ordem de dificuldade - do mais fácil para o mais difícil. Nós **não** levaremos em conta um exercício mais difícil completo se um mais fácil não estiver perfeitamente funcional.
- Usar uma função proibida é considerado trapaça. Trapaceiros levam -42, e esta nota não é negociável.
- Você somente deverá enviar uma função `main()` se nós pedirmos um **programa**.
- A Moulinette compila com as flags: `-Wall -Wextra -Werror`, e usa `cc`.
- Se o seu programa não compila, você receberá um 0.

- Você não pode deixar **nenhum** arquivo adicional na sua pasta além dos especificados no subject.
- Tem uma dúvida? Pergunte para o seu parceiro da direita. Ou então tente o seu parceiro da esquerda.
- Seu guia de referência chama-se `Google / man / a Internet /`
- Verifique o fórum "C Piscine" na intranet, ou o servidor do Discord da Piscina.
- Examine os exemplos atentamente. Pode haver detalhes neles que não estão explícitos no subject...
- Por Odin, por Thor! Use seu cérebro!!!



Não se esqueça de adicionar o cabeçalho padrão da 42 em cada um dos seus arquivos `.c/.h`. A norminette verifica se ele está lá!



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

Capítulo II

Preâmbulo

Veja a seguir uma discussão extraída da série Silicon Valley:

- Por que não usar Vim em vez de Emacs? (RISOS)
- Eu uso Vim em vez de Emacs.
- Deus, nos ajude! Está bem. Quer saber? Eu acho que isso não vai dar certo. Desculpa. Sabe, como vamos trazer filhos ao mundo com eles sabendo disso? Não seria justo com eles, você não acha?
- Filhos? Nós ainda nem transamos.
- E adivinha só, agora nunca vamos transar. Porque eu não vou ficar com alguém que usa espaços em vez de tabs.
- Richard! (APERTA A BARRA DE ESPAÇO VÁRIAS VEZES)
- Nossa. Tá bom. Adeus.
- Um tab economiza oito espaços! - (BATE A PORTA) - (ESTRONDO)

. . .

(RICHARD GEME DE DOR)

- Meu Deus! Richard, o que aconteceu?
- Eu tentei descer a escada oito degraus de uma vez só. Mas estou bem.
- A gente se vê, Richard.
- Só queria provar o meu argumento.

Felizmente, você não precisa utilizar emacs e sua barra de espaço para completar os exercícios.


Limite de hoje

O limite de validação para este projeto é 50%.

Fica a seu critério determinar qualo exercício permite que você atinja esse limite, e se você vai querer completar mais exercícios.

Capítulo III

Exercício 00 : ft_strcpy


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

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

```
char *ft_strcpy(char *dest, char *src);
```

Capítulo IV

Exercício 01 : ft_strncpy


	Exercício : 01
	ft_strncpy
	Pasta de entrega : <i>ex01/</i>
	Ficheiros para entregar : ft_strncpy.c
	Funções autorizadas : Nenhuma

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

```
char *ft_strncpy(char *dest, char *src, unsigned int n);
```


Capítulo V

Exercício 02 : ft_str_is_alpha

	Exercício : 02
ft_str_is_alpha	
Pasta de entrega : <i>ex02/</i>	
Ficheiros para entregar : ft_str_is_alpha.c	
Funções autorizadas : Nenhuma	


- Escreva uma função que retorne 1 se a string passada como parâmetro só contiver caracteres alfabéticos e retorne 0 se a função contiver outros tipos de caracteres.
- Ela deverá ser prototipada da seguinte maneira:

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

- Ela deverá retornar 1 se **str** for uma string vazia.

Capítulo VI

Exercício 03 : ft_str_is_numeric

	Exercício : 03
	ft_str_is_numeric
	Pasta de entrega : ex03/
	Ficheiros para entregar : ft_str_is_numeric.c
	Funções autorizadas : Nenhuma


- Escreva uma função que retorne 1 se a string passada como parâmetro só contiver números e retorne 0 se a função contiver outros tipos de caracteres.
- Ela deverá ser prototipada da seguinte maneira:

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

- Ela deverá retornar 1 se **str** for uma string vazia.

Capítulo VII

Exercício 04 : ft_str_is_lowercase

	Exercício : 04
	ft_str_is_lowercase
	Pasta de entrega : <i>ex04/</i>
	Ficheiros para entregar : ft_str_is_lowercase.c
	Funções autorizadas : Nenhuma


- Escreva uma função que retorne 1 se a string passada como parâmetro só contiver caracteres alfabéticos minúsculos e retorne 0 se a função contiver outros tipos de caracteres.
- Ela deverá ser prototipada da seguinte maneira:

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

- Ela deverá retornar 1 se **str** for uma string vazia.

Capítulo VIII

Exercício 05 : ft_str_is_uppercase

	Exercício : 05
ft_str_is_uppercase	
Pasta de entrega : <i>ex05/</i>	
Ficheiros para entregar : ft_str_is_uppercase.c	
Funções autorizadas : Nenhuma	


- Escreva uma função que retorne 1 se a string passada como parâmetro só contiver caracteres alfabéticos maiúsculos e retorne 0 se a função contiver outros tipos de caracteres.
- Ela deverá ser prototipada da seguinte maneira:

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

- Ela deverá retornar 1 se **str** for uma string vazia.

Capítulo IX

Exercício 06 : ft_str_is_printable

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


- Escreva uma função que retorne 1 se a string passada como parâmetro só contiver caracteres imprimíveis e retorne 0 se a função contiver outros tipos de caracteres.
- Ela deverá ser prototipada da seguinte maneira:

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

- Ela deverá retornar 1 se `str` for uma string vazia.

Capítulo X

Exercício 07 : ft_strupcase

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


- Escreva uma função que transforme todas as letras em maiúsculo.
- Ela deverá ser prototipada da seguinte maneira:

```
char *ft_strupcase(char *str);
```

- Ela deverá retornar **str**.

Capítulo XI

Exercício 08 : ft_strlowercase

	Exercício : 08
	ft_strlowercase
	Pasta de entrega : <i>ex08/</i>
	Ficheiros para entregar : ft_strlowercase.c
	Funções autorizadas : Nenhuma


- Escreva uma função que transforma todas as letras em minúsculo.
- Ela deverá ser prototipada da seguinte maneira:

```
char *ft_strlowercase(char *str);
```

- Ela deverá retornar **str**.

Capítulo XII

Exercício 09 : ft_strcapitalize

	Exercício : 09
ft_strcapitalize	
Pasta de entrega : <i>ex09/</i>	
Ficheiros para entregar : ft_strcapitalize.c	
Funções autorizadas : Nenhuma	

- Escreva uma função que deixe a primeira letra de cada palavra em maiúsculo e o resto da palavra em minúsculo.
- Uma palavra é uma string de caracteres alfanuméricos.
- Ela deverá ser prototipada da seguinte maneira:

```
char *ft_strcapitalize(char *str);
```

- Ela deverá retornar **str**.
- Por exemplo:


```
oi, tudo bem? 42palavras quarenta-e-duas; cinquenta+e+um
```

- Deve resultar:

```
Oi, Tudo Bem? 42palavras Quarenta-E-Duas; Cinquenta+E+Um
```


Capítulo XIII

Exercício 10 : ft_strlcpy


	Exercício : 10
ft_strlcpy	
Pasta de entrega : <i>ex10/</i>	
Ficheiros para entregar : ft_strlcpy.c	
Funções autorizadas : Nenhuma	

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

```
unsigned int ft_strlcpy(char *dest, char *src, unsigned int size);
```

Capítulo XIV

Exercício 11 : ft_putstr_non_printable

	Exercício : 11
ft_putstr_with_non_printable	
Pasta de entrega : <i>ex11/</i>	
Ficheiros para entregar : <i>ft_putstr_non_printable.c</i>	
Funções autorizadas : <i>write</i>	

- Escreva uma função que exiba uma string de caracteres na tela. Se essa string contiver caracteres não imprimíveis, eles devem ser mostrados na forma hexadecimal (em minúsculo) precedidos por uma contrabarra (ou barra inversa).
- Por exemplo, com este parâmetro:

```
Oi\nvoce esta bem?
```

- A função deverá mostrar:


```
Oi\0avoce esta bem?
```

- Ela deverá ser prototipada da seguinte maneira:

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

Capítulo XV

Exercício 12 : ft_print_memory

	Exercício : 12
	ft_print_memory
	Pasta de entrega : <i>ex12/</i>
	Ficheiros para entregar : <code>ft_print_memory.c</code>
	Funções autorizadas : <code>write</code>

- Escreva uma função que exiba uma zona de memória na tela.
- A exibição da zona de memória deve estar dividida em três "colunas" separadas por um espaço:
 - O endereço em hexadecimal do primeiro caractere da linha seguido por um ':':
 - O conteúdo em hexadecimal com um espaço nos dois caracteres e deve ser completado com espaços se necessário (veja o exemplo abaixo).
 - O conteúdo em caracteres imprimíveis.
- Se um caractere for não imprimível, deve ser substituído por um ponto.
- Cada linha deve administrar dezesseis caracteres.
- Se `size` for igual a 0, nada será mostrado.

- Exemplo:

```
$> ./ft_print_memory
000000010a161f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin
000000010a161f50: 6368 6573 090a 0963 0720 6573 7420 666f ches...c. est fo
000000010a161f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on
000000010a161f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.
000000010a161f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..
000000010a161f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000 ..lol.lol. .
$> ./ft_print_memory | cat -te
0000000107ff9f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin$
0000000107ff9f50: 6368 6573 090a 0963 0720 6573 7420 666f ches...c. est fo$
0000000107ff9f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on $
0000000107ff9f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.$
0000000107ff9f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..$
0000000107ff9f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000 ..lol.lol. .$.
$>
```

- Ela deverá ser prototipada da seguinte maneira:

```
void      *ft_print_memory(void *addr, unsigned int size);
```

- Ela deverá retornar addr.

Capítulo XVI

Entrega e avaliação entre pares

Entregue seu projeto em seu repositório `Git` como de costume. Somente o trabalho contido em seu repositório será avaliado durante a defesa. Não hesite em verificar mais de uma vez os nomes dos seus arquivos para ter certeza de que eles estão corretos.



Você deve submeter somente os arquivos solicitados pelo subject deste projeto.