



C Piscine

C 02

Preâmbulo: Este documento é o subject para o módulo C 02 da C Piscine na 42.

Versão: 6

Sumário

I	Instruções	2
II	Instruções de IA	4
III	Prefácio	7
IV	Exercício 00 : ft_strcpy	9
V	Exercício 01 : ft_strncpy	10
VI	Exercício 02 : ft_str_is_alpha	11
VII	Exercício 03 : ft_str_is_numeric	12
VIII	Exercício 04 : ft_str_is_lowercase	13
IX	Exercício 05 : ft_str_is_uppercase	14
X	Exercício 06 : ft_str_is_printable	15
XI	Exercício 07 : ft_strupcase	16
XII	Exercício 08 : ft_strlowcase	17
XIII	Exercício 09 : ft_strcapitalize	18
XIV	Exercício 10 : ft_strlcpy	19
XV	Exercício 11 : ft_putstr_non_printable	20
XVI	Exercício 12 : ft_print_memory	21
XVII	Submissão e avaliação por pares	23

Capítulo I

Instruções

- Somente esta página serve como sua referência, não confie em rumores.
- Cuidado! Este documento pode mudar antes da submissão.
- Certifique-se de que você tenha as permissões apropriadas em seus arquivos e diretórios.
- Você deve seguir os **procedimentos de submissão** para todos os seus exercícios.
- Seus exercícios serão verificados e corrigidos por seus colegas de classe.
- Além disso, seus exercícios serão avaliados por um programa chamado **Moulinette**.
- **Moulinette** é meticulosa e estrita em sua avaliação. Ela é totalmente automatizada, e não há como negociar com ela. Para evitar surpresas desagradáveis, seja o mais completo possível.
- **Moulinette** não é tolerante. Se seu código não aderir à Norma, ela não tentará entendê-lo. **Moulinette** utiliza um programa chamado **norminette** para verificar se seus arquivos estão em conformidade com a Norma. TL;DR: Submeter trabalho que não passa na verificação da **norminette** não faz sentido.
- Esses exercícios são organizados por ordem de dificuldade, do mais fácil para o mais difícil. Nós **não** consideraremos um exercício mais difícil concluído com sucesso se um exercício mais fácil não estiver totalmente funcional.
- Usar uma função proibida é considerado trapaça. Trapaceiros recebem uma nota de **-42**, o que é inegociável.
- Você só precisa submeter uma função **main()** se pedirmos especificamente um **programa**.
- **Moulinette** compila com as seguintes flags: **-Wall -Wextra -Werror**, usando **cc**.
- Se seu programa não compilar, você receberá uma nota **0**.

- Você **não pode** deixar **nenhum** arquivo adicional em seu diretório além daqueles especificados na tarefa.
- Tem uma pergunta? Pergunte ao colega à sua direita. Se não, tente o colega à sua esquerda.
- Seu guia de referência se chama **Google / man / Internet / ...**
- Verifique a seção "Piscine C" do fórum na intranet ou na Piscine no Slack.
- Examine cuidadosamente os exemplos. Eles podem conter detalhes cruciais que não são explicitamente declarados na tarefa...
- Por Odin, por Thor! Use seu cérebro!!!



O Norminette deve ser executado com a flag `-R CheckForbiddenSourceHeader`, que também será usada pela Moulinette.

Capítulo II

InSTRUÇÕES DE IA

● Contexto

A C Piscine é intensa. É seu primeiro grande desafio na 42 — um mergulho profundo na resolução de problemas, autonomia e comunidade.

Durante essa fase, seu objetivo principal é construir sua base — através do esforço, da repetição e, especialmente, da troca de **aprendizagem entre pares**.

Na era da IA, atalhos são fáceis de encontrar. No entanto, é importante considerar se o uso da IA está realmente ajudando você a crescer — ou simplesmente atrapalhando o desenvolvimento de habilidades reais.

A Piscine também é uma experiência humana — e, por enquanto, nada pode substituí-la. Nem mesmo a IA.

Para uma visão mais completa de nossa posição sobre a IA — como ferramenta de aprendizagem, como parte do currículo de TIC e como uma expectativa crescente no mercado de trabalho — consulte as perguntas frequentes disponíveis na intranet.

● Mensagem principal

- 👉 Construa bases sólidas sem atalhos.
- 👉 Desenvolva realmente habilidades técnicas e de poder.
- 👉 Experimente a verdadeira aprendizagem entre pares, comece a aprender como aprender e resolver novos problemas.
- 👉 A jornada de aprendizagem é mais importante que o resultado.
- 👉 Aprenda sobre os riscos associados à IA e desenvolva práticas de controle eficazes e contramedidas para evitar armadilhas comuns.

● **Regras para o aluno:**

- Você deve aplicar o raciocínio às tarefas atribuídas, especialmente antes de recorrer à IA.
- Você não deve pedir respostas diretas à IA.
- Você deve aprender sobre a abordagem global da 42 em relação à IA.

● **Resultados da fase:**

Dentro desta fase fundamental, você obterá os seguintes resultados:

- Obter bases adequadas em tecnologia e codificação.
- Saber por que e como a IA pode ser perigosa durante esta fase.

● **Comentários e exemplo:**

- Sim, sabemos que a IA existe — e sim, ela pode resolver seus projetos. Mas você está aqui para aprender, não para provar que a IA aprendeu. Não perca seu tempo (nem o nosso) apenas para demonstrar que a IA pode resolver o problema proposto.
- Aprender na 42 não é sobre saber a resposta — é sobre desenvolver a capacidade de encontrá-la. A IA lhe dá a resposta diretamente, mas isso impede você de construir seu próprio raciocínio. E o raciocínio leva tempo, esforço e envolve falhas. O caminho para o sucesso não deve ser fácil.
- Lembre-se de que durante os exames, a IA não está disponível — sem internet, sem smartphones, etc. Você perceberá rapidamente se confiou muito na IA em seu processo de aprendizagem.
- A aprendizagem entre pares o expõe a diferentes ideias e abordagens, melhorando suas habilidades interpessoais e sua capacidade de pensar de forma divergente. Isso é muito mais valioso do que apenas conversar com um bot. Portanto, não seja tímido — converse, faça perguntas e aprenda juntos!
- Sim, a IA fará parte do currículo — tanto como ferramenta de aprendizagem quanto como tópico em si. Você terá até a chance de construir seu próprio software de IA. Para saber mais sobre nossa abordagem crescente, consulte a documentação disponível na intranet.

✓ Boa prática:

Estou travado em um novo conceito. Pergunto a alguém próximo como ele abordou isso. Conversamos por 10 minutos — e de repente, clica. Entendi.

✗ Má prática:

Usei secretamente a IA, copiei um código que parece certo. Durante a avaliação entre pares, não consigo explicar nada. Eu falho. Durante o exame — sem IA — estou travado novamente. Eu falho.

Capítulo III

Prefácio

Aqui está um trecho de uma discussão na série *Silicon Valley*:

- Quero dizer, por que não usar Vim ao invés do Emacs? (CHUCKLES)
- Eu uso Vim o invés do Emacs.
- Oh, Deus, nos ajude! Ok, sabe de uma coisa? Eu simplesmente não acho que isso vai funcionar. Me desculpe. Quer dizer, tipo, vamos trazer crianças ao mundo com isso pairando sobre suas cabeças? Isso não é justo com elas, não acha?
- Crianças? Nós nem dormimos juntos.
- E sabe de uma coisa? Isso nunca vai acontecer agora, porque não há a mínima possibilidade de eu ficar com alguém que usa espaços em vez de tabulações.
- Richard! (PRESSIONA A BARRA DE ESPAÇO VÁRIAS VEZES)
- Nossa. Okay. Tchau.
- Uma tabulação economiza oito espaços! - (PORTA BATE) - (BATIDA)

. . .

(RICHARD GEME)

- Oh, my God! Richard, what happened?
- I just tried to go down the stairs eight steps at a time. I'm okay, though.
- See you around, Richard.
- Just making a point.

Esperamos que você não precise usar o emacs e sua barra de espaço para completar os exercícios a seguir.

Limite de hoje

O limite de validação para este projeto é de 50%.

Cabe a você determinar quais exercícios permitem que você alcance este limite e se você deseja completar exercícios adicionais.

Capítulo IV

Exercício 00 : ft_strdup

	Exercício : 00
	ft_strdup
	Pasta de entrega : <i>ex00/</i>
	Arquivos para entregar : ft_strdup.c
	Funções ou bibliotecas autorizadas : Nenhuma

- Reproduzir de forma idêntica o funcionamento da função **strcpy** (man strcpy).
- O protótipo deve ser:

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

Capítulo V

Exercício 01 : ft_strncpy

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

- Reproduzir de forma idêntica o funcionamento da função **strncpy** (man strncpy).
- O protótipo deve ser:

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

Capítulo VI

Exercício 02 : ft_str_is_alpha

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

- Crie uma função que retorne 1 se a string fornecida contiver apenas caracteres alfabéticos e 0 se contiver qualquer outro caractere.
- O protótipo deve ser:

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

- Deve retornar 1 se **str** estiver vazia.

Capítulo VII

Exercício 03 : ft_str_is_numeric

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

- Crie uma função que retorne 1 se a string fornecida contiver apenas dígitos e 0 se contiver qualquer outro caractere.
- O protótipo deve ser:

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

- Deve retornar 1 se **str** estiver vazia.

Capítulo VIII

Exercício 04 : ft_str_is_lowercase

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

- Crie uma função que retorne 1 se a string fornecida contiver apenas caracteres alfabéticos minúsculos e 0 se contiver qualquer outro caractere.
- O protótipo deve ser:

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

- Deve retornar 1 se **str** estiver vazia.

Capítulo IX

Exercício 05 : ft_str_is_uppercase

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

- Crie uma função que retorne 1 se a string fornecida contiver apenas caracteres alfabéticos maiúsculos e 0 se contiver qualquer outro caractere.
- O protótipo deve ser:

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

- Deve retornar 1 se **str** estiver vazia.

Capítulo X

Exercício 06 : ft_str_is_printable

	Exercício : 06
	ft_str_is_printable
	Pasta de entrega : <i>ex06/</i>
	Arquivos para entregar : ft_str_is_printable.c
	Funções ou bibliotecas autorizadas : Nenhuma

- Crie uma função que retorne 1 se a string fornecida contiver apenas caracteres imprimíveis e 0 se contiver qualquer outro caractere.
- O protótipo deve ser:

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

- Deve retornar 1 se **str** estiver vazia.

Capítulo XI

Exercício 07 : ft_strdupcase

	Exercício : 07
	ft_strdupcase
	Pasta de entrega : <i>ex07/</i>
	Arquivos para entregar : ft_strdupcase.c
	Funções ou bibliotecas autorizadas : Nenhuma

- Crie uma função que converta cada letra para maiúscula.
- O protótipo deve ser:

```
char *ft_strdupcase(char *str);
```

- Deve retornar **str**.

Capítulo XII

Exercício 08 : ft_strlowlcase

	Exercício : 08
	ft_strlowlcase
	Pasta de entrega : <i>ex08/</i>
	Arquivos para entregar : <i>ft_strlowlcase.c</i>
	Funções ou bibliotecas autorizadas : Nenhuma

- Crie uma função que converta cada letra para minúscula.
- O protótipo deve ser:

```
char *ft_strlowlcase(char *str);
```

- Deve retornar **str**.

Capítulo XIII

Exercício 09 : ft_strcapitalize

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

- Crie uma função que capitalize a primeira letra de cada palavra e converta todas as outras letras para minúsculas.
- Uma palavra é uma sequência de caracteres alfanuméricos.
- O protótipo deve ser:

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

- Deve retornar **str**.
- Por exemplo:

```
hi, how are you? 42words forty-two; fifty+and+one
```

- Torna-se:

```
Hi, How Are You? 42words Forty-Two; Fifty+And+One
```

Capítulo XIV

Exercício 10 : ft_strlcpy

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

- Reproduzir de forma idêntica o funcionamento da função **strlcpy** (man strlcpy).
- O protótipo deve ser:

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

Capítulo XV

Exercício 11 : **ft_putstr_non_printable**

	Exercício : 11
	<u>ft_putstr_non_printable</u>
	Pasta de entrega : <i>ex11/</i>
	Arquivos para entregar : ft_putstr_non_printable.c
	Funções ou bibliotecas autorizadas : write

- Crie uma função que exiba uma string de caracteres na tela. Se esta string conter caracteres não imprimíveis, eles devem ser exibidos como valores hexadecimais minúsculos, precedidos por uma barra invertida.
- Por exemplo:

```
Hello\nHow are you?
```

- A função deve exibir:

```
Hello\0aHow are you?
```

- O protótipo deve ser:

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

Capítulo XVI

Exercício 12 : ft_print_memory

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

- Crie uma função que exiba uma área de memória na tela.
- A exibição desta área de memória deve ser dividida em três "colunas", separadas por um espaço:
 - O endereço hexadecimal do primeiro caractere na linha, seguido por um ":".
 - O conteúdo em hexadecimal, com um espaço a cada dois caracteres, e preenchido com espaços se necessário (veja o exemplo abaixo).
 - O conteúdo em caracteres imprimíveis.
- Se um caractere não for imprimível, ele deve ser substituído por um ponto.
- Cada linha deve exibir dezesseis caracteres.
- Se **size** for igual a 0, nada deve ser exibido.

- 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. .$
```

- O protótipo deve ser:

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

- Deve retornar **addr**.

Capítulo XVII

Submissão e avaliação por pares

Envie seu trabalho em seu repositório `Git` como de costume. Apenas o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes dos seus arquivos para garantir que estejam corretos.



Você deve enviar apenas os arquivos exigidos pelo tema deste projeto.