



Piscina C

C 09

Sumário: Este documento é o tema do módulo C 09 da Piscina C da 42.

Versão: 3.3

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : libft	5
IV	Exercício 01 : Makefile	6
V	Exercício 02 : ft_split	8
VI	Submissão e Avaliação	9

Capítulo I

Instruções

- Somente este documento servirá de referência; não confie nos boatos.
- Leia bem o enunciado antes de entregar os seus exercícios. A qualquer momento pode haver alterações.
- Tenha atenção aos direitos dos seus ficheiros e 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, seja rigoroso!
- A Moulinette não tem uma mente muito aberta: não tenta entender código que não respeita a Norma. A Moulinette utiliza o programa `norminette` para verificar a norma dos ficheiros. Seria uma tontice entregar código que não passa pela `norminette`...
- 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.
- A utilização de qualquer função proibida é um caso de fraude. Qualquer fraude é punida com nota de -42.
- Deve entregar uma função `main()` se for pedido um programa.
- A Moulinette compila com as flags `-Wall -Wextra -Werror`, e utiliza `cc`.
- Se o seu programa não compila, terá 0.
- Não deve deixar no repositório de entrega nenhum outro ficheiro além daqueles explicitamente especificados pelo enunciado 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...
- 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

Diálogo do filme O Grande Lebowski:

Walter: Desculpa, Smokey. Você pisou na linha: falta.

Smokey: Mentira! Marca oito, Cara...

Walter: Me desculpa. Marca zero. Próxima rodada.

Smokey: Deixa de ser chato, Walter. Marca oito, Cara...

Walter: Smokey, não estamos no Vietnã. Estamos no boliche. A gente joga com regras...

O Cara: Vai, Walter, não enche. O pé dele escorregou um pouquinho na linha... É só um jogo!

Walter: Sim. E é um jogo de campeonato, decide quem vai pra próxima fase, não é?

Smokey: É, mas eu não...

Walter: Não é verdade?

Smokey: Sim, mas eu não pisei... Vai! Cara, marca oito.

Walter: [saca sua arma] Smokey, meu amigo, se você nunca sentiu dor, você vai sentir...

Cara: Walter, não seja idiota!

Walter [ameaçador] Vai, marca um oito que você vai ver!

Smokey: Eu não...

Walter: Você vai sentir dor, Smokey!

Smokey: Cara? É o seu parceiro...

Walter: O mundo todo tá FICANDO LOUCO? [Ele se levanta] EU SOU O ÚNICO QUE LIGA PRAS REGRAS? MARCA ZERO!

Cara: Eles vão chamar a polícia, Walter, guarda essa...

Walter: [aponta a arma para Smokey] MARCA ZERO!

Cara: Guarda a arma, Walter...

Smokey: ...

Cara: Walter...


Walter: [engatilha a arma] VOCÊ ACHA QUE EU TÔ BRINCANDO? MARCA ZERO!

Smokey: Tá bom, marquei zero... Tá feliz? Seu louco!

Walter: [se acalma] ... É um jogo de campeonato.

Capítulo III

Exercício 00 : libft

	Exercício : 00
libft	
Pasta de entrega : <i>ex00/</i>	
Ficheiros para entregar : <code>libft_creator.sh</code> , <code>ft_putchar.c</code> , <code>ft_swap.c</code> , <code>ft_putstr.c</code> , <code>ft_strlen.c</code> , <code>ft_strcmp.c</code>	
Funções autorizadas : <code>write</code>	

- Crie a sua biblioteca `ft`. Ela vai chamar-se `libft.a`.
- Um script shell chamado `libft_creator.sh` vai compilar adequadamente os ficheiros de código-fonte e criar a sua biblioteca.
- Essa biblioteca deve conter todas as seguintes funções:


```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- Nós vamos executar o seguinte comando:

```
sh libft_creator.sh
```

Capítulo IV

Exercício 01 : Makefile

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

- Escreva o **Makefile** que compile a biblioteca **libft.a**.
- O seu **makefile** deve mostrar claramente cada comando que efetua.
- O seu **makefile** não deve efetuar comandos inúteis.
- O **Makefile** vai procurar os ficheiros de código-fonte na pasta **^srcs^**.
- Esses ficheiros de código-fonte serão: **ft_putchar.c**, **ft_swap.c**, **ft_putstr.c**, **ft_strlen.c**, **ft_strcmp.c**
- O **Makefile** vai procurar os ficheiros de cabeçalhos na pasta **"includes"**.
- Esses ficheiros de cabeçalhos serão: **ft.h**
- Ele deverá compilar seus ficheiros **c** utilizando **cc** e as opções de sinalizações **-Wall** **-Wextra** **-Werror** nessa ordem.
- A **lib** estará na origem do exercício.
- Os ficheiros **.o** deverão estar ao lado do seu respectivo arquivo **.c**.
- O **Makefile** deverá também implementar as regras **clean**, **fclean**, **re**, **all** e, claro, **libft.a**.
- Executar apenas **make** deve ser equivalente a **make all**.


- A regra `all` deverá agir como `make libft.a`.
- A regra `clean` deverá eliminar todos os ficheiros temporários gerados.
- A regra `fclean` é o equivalente a um `make clean` e também apaga os binários criados no `make all`.
- A regra `re` é o equivalente a um `make fclean` seguido de um `make all`.
- Seu makefile não deve recompilar os ficheiros desnecessariamente.
- Nós vamos pegar só no seu Makefile e testar com os nossos ficheiros.



Cuidado com os wildcards!

Capítulo V

Exercício 02 : ft_split

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

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

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

Capítulo VI

Submissão e Avaliação

Entrega a tua tarefa no teu repositório `Git` como habitualmente. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em verificar duas vezes os nomes dos teus ficheiros para garantir que estão corretos.



Deves devolver apenas os ficheiros solicitados pelo enunciado deste projeto.