



Piscina C

C 11

Sumário: Este documento é o tema do módulo C 11 da Piscina C da 42.

Versão: 6.2

Conteúdo

I	Instruções	2
II	Preâmbulo	4
III	Exercício 00 : ft_foreach	6
IV	Exercice 01 : ft_map	7
V	Exercício 02 : ft_any	8
VI	Exercício 03 : ft_count_if	9
VII	Exercício 04 : ft_is_sort	10
VIII	Exercício 05 : do-op	11
IX	Exercício 06 : ft_sort_string_tab	13
X	Exercício 07 : ft_advanced_sort_string_tab	14
XI	Entrega e avaliação entre pares	15

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 de 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 pedirmos um programa.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza `cc`.
- 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 na intranet, assim como no Discord 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.

Capítulo II

Preâmbulo

Aqui vai uma estorinha:

(1982, Califórnia) Larry Walters, de Los Angeles, é um dos poucos a disputar o Prêmio Darwin e viver para contar a história. "Realizei meu sonho de 20 anos", disse Walters, ex-motorista de caminhão de uma empresa que faz comerciais de TV. "Vou ficar no chão. Provei que a coisa funciona." O sonho de infância de Larry era voar. Mas o destino conspirou para afastá-lo de seu sonho. Ele ingressou na Força Aérea, mas sua visão deficiente o desqualificou para o cargo de piloto. Depois que ele foi dispensado do serviço militar, ele se sentou em seu quintal observando os jatos voarem sobre sua casa.

Ele traçou seu esquema de balão meteorológico enquanto estava sentado do lado de fora em sua cadeira de grama "extremamente confortável" da Sears. Ele comprou 45 balões meteorológicos de uma loja de excedentes do Exército-Marinha, amarrou-os em sua cadeira de jardim amarrada (apelidada de Inspiration I) e encheu os balões de um metro de diâmetro com hélio. Então, armado com alguns sanduíches, Miller Lite e uma pistola de cola, ele se amarrou em sua cadeira de jardim. Ele imaginou que atiraria para estourar alguns dos balões quando chegasse a hora de descer.

Larry planejou cortar a âncora e flutuar preguiçosamente a uma altura de cerca de 30 pés do quintal, onde desfrutaria de algumas horas de vôo antes de voltar para baixo. Mas as coisas não funcionaram exatamente como Larry planejou.

Quando seus amigos cortaram a corda que prendia a cadeira de jardim ao seu jipe, ele flutuou preguiçosamente até 30 pés. Em vez disso, ele disparou no céu de Los Angeles como se tivesse sido disparado por um canhão, puxado pela elevação de 45 balões de hélio contendo 33 pés cúbicos de hélio cada.

Ele não nivelou a 100 pés, nem nivelou a 1000 pés. Depois de subir e subir, ele nivelou a 16.000 pés.

Naquela altura, ele sentiu que não poderia arriscar atirar em nenhum dos balões, para não desequilibrar a carga e realmente se encontrar em apuros. Então ele ficou lá, morrendo de frio e assustado com sua cerveja e sanduíches, por mais de 14 horas. Ele atravessou o corredor de aproximação principal do LAX, onde pilotos assustados da Trans World Airlines e da Delta Airlines comunicaram pelo rádio a estranha visão.


Por fim, ele tomou coragem para atirar em alguns balões e desceu lentamente. As cordas penduradas se enredaram e ficaram presas em uma linha de energia, deixando um bairro de Long Beach escuro por 20 minutos. Larry escalou para um local seguro, onde foi preso por membros do LAPD que esperavam. Enquanto ele era levado algemado, um repórter enviado para cobrir o ousado resgate perguntou por que ele havia feito isso. Larry respondeu com indiferença: "Um homem não pode simplesmente ficar sentado".

A Federal Aviation Administration não achou graça. O inspetor de segurança Neal Savoy "Sabemos que ele quebrou alguma parte da Lei Federal de Aviação e, assim que decidirmos, uma acusação será feita."

A moral da estória pe que Larry Walters deveria ter ficado sentado em sua cadeira e aprendido C...

Capítulo III

Exercício 00 : ft_foreach

	Exercício : 00
	ft_foreach
	Pasta de entrega : ex00/
	Ficheiros para entregar : ft_foreach.c
	Funções autorizadas : Nenhuma

- Escreva uma função `ft_foreach` que, para uma matriz de inteiros dada, aplique uma função sobre todos os elementos dessa matriz. Essa função será aplicada na ordem da matriz.
- A função deverá ser prototipada da seguinte maneira:


```
void ft_foreach(int *tab, int length, void(*f)(int));
```

- Por exemplo, a função `ft_foreach` poderá ser chamada da seguinte forma para mostrar o conjunto de inteiros da matriz:

```
ft_foreach(tab, 1337, &ft_putnbr);
```

Capítulo IV

Exercice 01 : ft_map


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

- Escreva uma função **ft_map** que, para uma matriz de inteiros dada, aplicará uma função sobre todos os elementos dessa matriz (em sequência) e retornará uma matriz de todos os valores de retorno.
- Essa função será aplicada na ordem da matriz.
- A função deverá ser prototipada da seguinte maneira:

```
int      *ft_map(int *tab, int length, int(*f)(int));
```


Capítulo V

Exercício 02 : ft_any

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


- Escreva uma função **ft_any** que retornará 1 se pelo menos um elemento da matriz retornar algo diferente de 0 ao ser passado para a função **f**. Caso contrário, ela deve retornar 0.
- Essa função será aplicada na ordem da matriz.
- A função deverá ser prototipada da seguinte maneira:

```
int ft_any(char **tab, int(*f)(char*));
```

- A matriz terminará com um ponteiro nulo.

Capítulo VI

Exercício 03 : ft_count_if


	Exercício : 03
	ft_count_if
	Pasta de entrega : <i>ex03/</i>
	Ficheiros para entregar : <code>ft_count_if.c</code>
	Funções autorizadas : Nenhuma

- Escreva uma função `ft_count_if` que retornará o número de elementos da matriz que, ao serem passados para a função `f`, não retornam 0.
- Essa função será aplicada na ordem da matriz.
- A função deverá ser prototipada da seguinte maneira:

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

Capítulo VII

Exercício 04 : ft_is_sort


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

- Escreva uma função `ft_is_sort` que retornará 1 se a matriz estiver ordenada e 0 no caso contrário.
- A função passada como parâmetro retornará um inteiro negativo se o primeiro argumento for inferior ao segundo, 0 se forem iguais e um inteiro positivo em caso contrário.
- A função deverá ser prototipada da seguinte maneira:

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

Capítulo VIII

Exercício 05 : do-op

	Exercício : 05
do-op	
Pasta de entrega : <i>ex05/</i>	
Ficheiros para entregar : Todos os arquivos necessários para o seu programa	
Funções autorizadas : <code>write</code>	

- Escreva um programa chamado `do-op`.
- O programa deverá ser executado com três argumentos: `do-op valor1 operador valor2`
- Exemplo:

```
$>./do-op 42 "+" 21
63
$>
```

- Você deve utilizar uma matriz de ponteiros para função a fim de chamar a função correspondente a um `operador`.
- Em caso de operador desconhecido, seu programa deverá mostrar 0.
- Se o número de argumentos não estiver correto, `do-op` não mostra nada.
- Seu programa deve aceitar e mostrar o resultado com os seguintes operadores: `'+' '-' '/' '*'` e `'%'`
- Em caso de divisão por 0, seu programa deve mostrar:

```
Stop : division by zero
```

- Em caso de módulo por 0, seu programa deve mostrar:


```
Stop : modulo by zero
```

- Veja um exemplo de testes da Moulinette :

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

Capítulo IX

Exercício 06 : ft_sort_string_tab

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


- Escreva a função `ft_sort_string_tab` que classifica por ordem `ascii` as strings em `tab`.
- `tab` terminará com um ponteiro nulo
- A classificação será realizada trocando os ponteiros da matriz.
- Ela deverá ser prototipada da seguinte maneira:

```
void ft_sort_string_tab(char **tab);
```

Capítulo X

Exercício 07 :

ft_advanced_sort_string_tab

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

- Escreva a função `ft_advanced_sort_string_tab` que classifica em função do retorno da função passada como parâmetro
- A classificação será realizada trocando os ponteiros da matriz.
- `tab` terminará com um ponteiro nulo
- Ela deverá ser prototipada da seguinte maneira:

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



Um chamado à `ft_advanced_sort_string_tab()` com `emft_strcmp` como segundo parâmetro dará o mesmo resultado que `ft_sort_string_tab()`.

Capítulo XI

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.