



C Piscina

Rush 00

Resumo: Este documento é o assunto do Rush00 da C Piscine @ 42.

Versão: 5.2

Conteúdo

I	Instruções	2
II	Prefácio	4
III	assunto principal	6
IV	Corrida 00	8
V	Rush 01	10
VI	Rush 02	11
VII	Corrida 03	12
VIII	Corrida 04	13
IX	Submissão e avaliação por pares	14

Capítulo I

Instruções

- O grupo SERÁ cadastrado para defesa automaticamente.
- Não cancele, você não receberá um segundo.
- Qualquer pergunta sobre o assunto complicaria o assunto.
- Você deve seguir os procedimentos de submissão para todos os seus exercícios.
- Este assunto pode mudar até uma hora antes do envio.
- O programa deve compilar com os seguintes flags: -Wall -Wextra -Werror; e usa cc.
- Se o seu programa não compilar, você obterá 0.
- Seu programa deve ser escrito de acordo com a Norma. Se você tiver arquivos/funções de bônus, eles serão incluídos na verificação da norma e você receberá um 0 se houver um erro de norma dentro.
- Você terá que lidar com os erros de forma coerente. Sinta-se à vontade para imprimir uma mensagem de erro ou simplesmente devolver o controle ao usuário.
- Os exercícios Rushes devem ser realizados em grupo de 2, 3 ou 4.
- O número de corrida obrigatório para sua equipe seguirá esta regra:
Índice alfabético da primeira letra do login do líder da equipe (de 1 a 26) módulo 5.
- Você deve, portanto, fazer o projeto com a equipe imposta e aparecer no seu slot de defesa, com todos os seus companheiros de equipe.
- Seu projeto deve estar pronto antes da defesa. A finalidade da defesa é para você apresentar e explicar o seu trabalho.
- Cada membro do seu grupo deve estar totalmente ciente dos trabalhos do projeto. Se você optar por dividir a carga de trabalho, certifique-se de que todos entendam o que todos fizeram. Durante a defesa, serão feitas perguntas e a nota final será baseada nas piores explicações.

- Nem é preciso dizer, mas reunir o grupo é sua responsabilidade. Você tem todos os meios para entrar em contato com seus companheiros de equipe: telefone, e-mail, pombo-correio, espiritismo, etc. Portanto, não se preocupe em inventar desculpas. A vida não é justa, é assim mesmo.
- No entanto, se você realmente tentou de tudo, um de seus companheiros continua inacessível: faça o projeto de qualquer maneira e tentaremos ver o que podemos fazer sobre isso durante a defesa. Mesmo que o líder do grupo esteja ausente, você ainda terá acesso ao diretório de envio.
- Se você quiser pontos de bônus, você pode enviar outras disciplinas ou usar o programa argumentos para testar sua função.



Certifique-se de que a disciplina originalmente atribuída ao seu grupo funcione perfeitamente antes de considerar os bônus: Se uma disciplina de bônus funcionar, mas a original for reprovada nos testes, você receberá 0.

Capítulo II

Prefácio

Aqui estão as letras de um famoso programa de TV para todos:

[Verso 1]

Eu quero ser o melhor
Como ninguém jamais foi

Pegá-los é meu verdadeiro teste
Treiná-los é minha causa

Eu viajarei pela terra

Procurando por toda parte
Cada pokémon para entender
O poder que está dentro

[Coro]

Pokémon! Tenho que pegá-los todos! É você e eu, eu sei que é o meu
destino, Pokémon! Oh você é meu melhor amigo Em um mundo,
devemos defender Pokémon! Um coração tão verdadeiro

Nossa coragem nos levará adiante,

Você me ensina e eu te ensino, Pokémon! Tenho
que pegar todos

[Coro]

Cada desafio do caminho Com coragem eu
enfrentarei.

Eu batalharei todos os dias Para
reivindicar meu lugar de direito.

Vem comigo, é a

hora certa, melhor time não
tem.

De braços dados venceremos a luta!

Sempre foi nosso sonho!

C Piscina


Rush 00

[Coro]

Eu poderia apostar que você estava cantando agora, mas isso não importa no momento. E esse assunto não é relacionado ao Pocket Monster por sinal...

Capítulo III

Assunto principal

	Exercício 00
	Rush0X
	Diretório de entrega: ex00/
	Arquivos a serem entregues: main.c, ft_putchar.c, rush0X.c
	Funções permitidas: escrever

- Arquivos a serem enviados: main.c, ft_putchar.c e seu rush0X.c, '0X' representa o número do rush. Por exemplo rush00.c.
- Esses três arquivos serão compilados juntos.
- Seu arquivo ft_putchar.c deve incluir a função ft_putchar.
- Exemplo de main.c:

```
int principal()
{
    pressa(5, 5);
    retornar (0);
}
```

- Você deve, portanto, escrever a função rush tomando como argumentos duas variáveis do tipo int, denominadas respectivamente x e y. Não há necessidade de dizer que esta função deve estar no arquivo rush0X.c.
- Sua função rush deve exibir (na tela) um retângulo de x caracteres para largura e y caracteres para comprimento.
- Sua função nunca deve travar ou entrar em loop indefinidamente.
- Seu principal será modificado durante a defesa, para verificar se você fez tudo o que deveria. Aqui está um exemplo de teste que faremos:

```
int    principal()
{
    pressa(123, 42);
    retornar (0);
}
```


Capítulo IV

Rush 00

- rush(5,3) deve exibir:

```
$> ./a.out  
o--o  
|  |  
o--o  
$>
```

- rush(5, 1) deve exibir:

```
$> ./a.out  
o--o  
$>
```

- rush(1, 1) deve exibir:

```
$> ./a.out  
o  
$>
```

- rush(1, 5) deve exibir:

```
$> ./a.out  
o  
.  
.  
.  
.  
o  
$>
```

- rush(4, 4) deve exhibir:

```
$> ./a.out
```

```
o-o
```

```
  "  "
```

```
o-o
```

```
$>
```

Capítulo V

Corrida 01

- `rush(5,3)` deve exibir:

```
$> ./a.out /  
***\  
**  
\\**/ $>
```

- `rush(5, 1)` deve exibir:

```
$> ./a.out /  
***\  
$>
```

- `rush(1, 1)` deve exibir:

```
$> ./a.out /  
$>
```

- `rush(1, 5)` deve exibir:

```
$> ./a.out /  
*  
*  
*  
\  
$>
```

- `rush(4, 4)` deve exibir:

```
$> ./a.out /  
**\  
**  
**  
\\**/  
$>
```

Capítulo VI

Corrida 02

- rush(5,3) deve exibir:

```
$> ./a.out  
ABBBA BB  
  
CBBBC$>
```

- rush(5, 1) deve exibir:

```
$> ./a.out  
ABBBA $>
```

- rush(1, 1) deve exibir:

```
$> ./a.out A  
$>
```

- rush(1, 5) deve exibir:

```
$> ./a.out A  
  
B  
B  
B  
  
C$>
```

- rush(4, 4) deve exibir:

```
$> ./a.out  
ABBA  
bb  
bb  
CBBBC  
$>
```

Capítulo VII

Corrida 03

- `rush(5,3)` deve exibir:

```
$> ./a.out  
ABBBC BB  
  
ABBBC  
$>
```

- `rush(5, 1)` deve exibir:

```
$> ./a.out  
ABBBC $>
```

- `rush(1, 1)` deve exibir:

```
$> ./a.out A  
$>
```

- `rush(1, 5)` deve exibir:

```
$> ./a.out A  
  
B  
B  
B  
  
A $>
```

- `rush(4, 4)` deve exibir:

```
$> ./a.out  
ABBC  
bb  
bb  
ABBC  
$>
```

Capítulo VIII

Corrida 04

- `rush(5,3)` deve exibir:

```
$> ./a.out  
ABBBC BB  
  
CBBBA$>
```

- `rush(5, 1)` deve exibir:

```
$> ./a.out  
ABBBC $>
```

- `rush(1, 1)` deve exibir:

```
$> ./a.out A  
$>
```

- `rush(1, 5)` deve exibir:

```
$> ./a.out A  
  
B  
B  
B  
  
C$>
```

- `rush(4, 4)` deve exibir:

```
$> ./a.out  
ABBC  
bb  
bb  
CBBA  
$>
```

Capítulo IX

Submissão e avaliação por pares

Entregue sua atribuição 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 de seus arquivos para garantir que estejam corretos.

Como essas atribuições não são verificadas por um programa, fique à vontade para organizar seus arquivos como quiser, desde que entregue os arquivos obrigatórios e cumpra os requisitos.



Você precisa devolver apenas os arquivos solicitados pelo assunto deste projeto.