

Piscina C Rush 00

 $Sum\'ario:\ ESTE\ documento\ \'e\ o\ enunciado\ do\ m\'odulo\ Rush\ 00\ da\ Piscina\ C\ da\ 42.$

Versão: 5.2

Conteúdo

1	Instruções	2
II	Preâmbulo	4
III	Enunciado Principal	6
IV	Rush 00	8
\mathbf{V}	Rush 01	10
\mathbf{VI}	Rush 02	11
VII	Rush 03	12
VIII	Rush 04	13
IX	Submissão e avaliação	14

Capítulo I

Instruções

- O grupo será automaticamente inscrito para defesa.
- No caso de ser cancelada, não terão direito a outra.
- Qualquer questão acerca do enunciado tem a consequência de o complicar.
- Deverá seguir o procedimento de entrega para o enunciado.
- O enunciado pode mudar até uma hora antes da entrega.
- O programa deverá compilar com as seguintes *flags*: -Wall -Wextra -Werror; e utiliza cc.
- Se o programa não compila, o grupo terá 0.
- O programa deve ser escrito de acordo com a Norma. Se tiveres arquivos/funções de bônus, eles estão incluídos na verificação da norma e receberás um 0 se houver um erro de norma dentro deles.
- Terás que lidar com os erros de forma coerente. Sintam-se à vontade para imprimir uma mensagem de erro ou simplesmente devolver o controle ao utilizador.
- Os exercicios do Rush devem ser realizados por grupos de 2, 3 ou 4 pessoas.
- O número de rush obrigatório do seu grupo seguirá esta regra:
 O índice alfabético da primeira letra do login do líder da equipa (de 1 a 26) módulo
 5.
- Deverá, portanto, executar o enunciado indicado com os seus parceiros impostos e apresentar-se para a defesa do projeto na hora marcada com todos os seus parceiros.
- O projeto deverá estar finalizado pela altura da defesa. O propósito da defesa é para que todos possam apresentar e explicar o projeto em todo o seu detalhe.

Piscina C

• Cada membro do grupo deve estar perfeitamente consciente do trabalho realizado. No caso de terem dividido o trabalho, certifiquem-se que todos perceberam o trabalho de cada um. Durante a defesa, serão feitas várias questões; a nota do grupo será baseada nas piores explicações.

Rush 00

- O contacto entre o grupo é da vossa responsabilidade. Têm todos os meios para entrar em contacto com os parceiros: telefone, carta, pombo correio, bola de cristal, etc. Não há necessidade de desculpas quanto ao funcionamento do grupo. A vida nem sempre é justa; é o que é.
- No entanto, se depois de <u>tentar realmente tudo</u>, um dos seus parceiros ainda estiver incomunicável: faça o rush de qualquer forma, e arranjaremos uma solução na defesa. Incluindo o caso da ausência do líder do grupo: todos possuem acesso ao repositório.
- Bom trabalho a todos.
- Se quiserem pontos bónus, podem submeter outros enunciados ou serem capazes de usar argumentos no programa para testar a vossa função.



Deve fazer o enunciado obrigatório de forma <u>perfeita</u> para apresentar os enunciados bónus: Se um enunciado bónus funcionar mas o original falhar os testes, o grupo terá 0.

Capítulo II

Preâmbulo

Here are the lyrics of a famous TV show for everyone:

[Verse 1]
I wanna be the very best
Like no one ever was
To catch them is my real test
To train them is my cause

I will travel across the land Searching far and wide Each pokemon to understand The power that's inside

[Chorus]

Pokemon! Gotta catch 'em all! It's you and me I know it's my destiny,
Pokemon! Oh you're my best friend
In a world, we must defend
Pokemon! A heart so true
Our courage will pull us through,

You teach me and I'll teach you, Pokemon! Gotta catch'em all

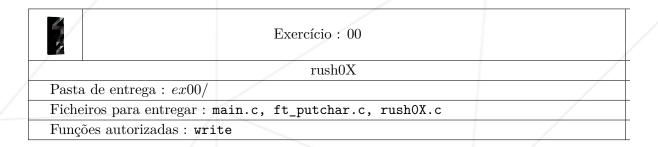
[Chorus]

Every challenge along the way
With courage I will face.
I will battle every day
To claim my rightful place.
Come with me,
The time is right,
There's no better team.
Arm in arm we'll win the fight!
It's always been our dream!

Piscina C		Rush 00
		1
[Chorus]		
		C. (1. (A 1
this subject is not related to F	ng right now, but it doesn't matter Pocket Monster by the way	for the moment. And
	5	
	9	

Capítulo III

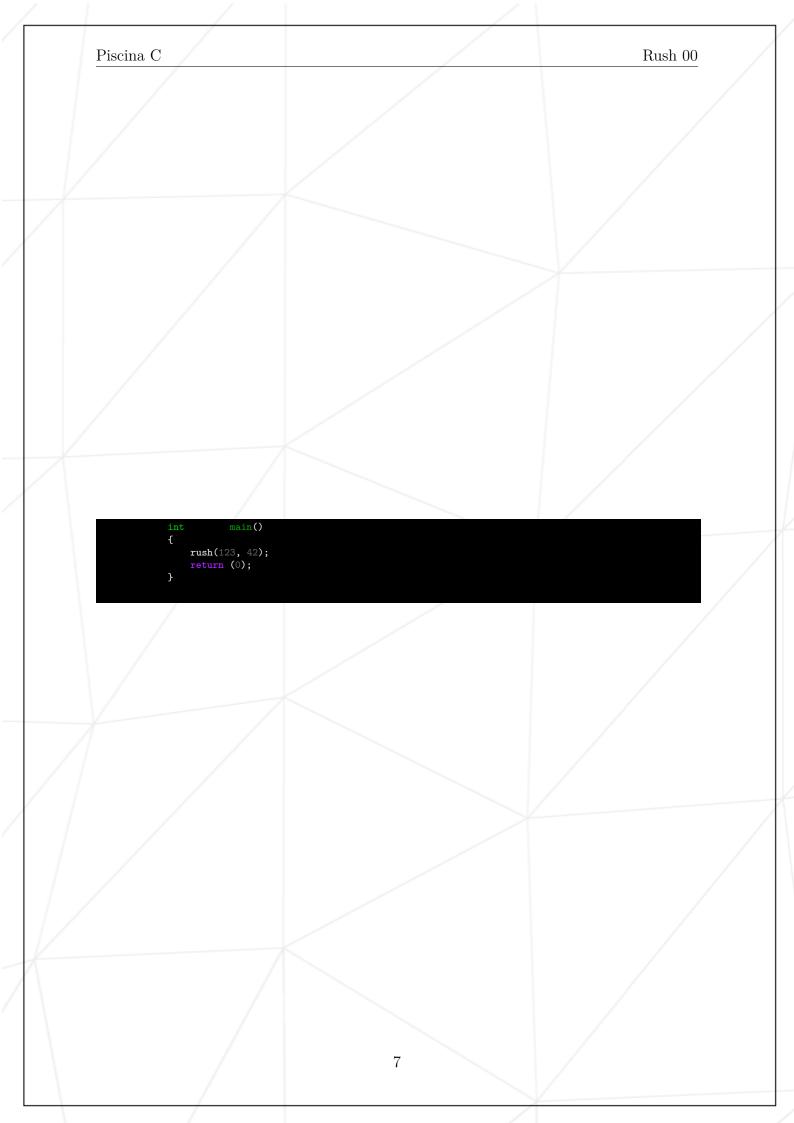
Enunciado Principal



- Os arquivos que devem ser entregues são: main.c, ft_putchar.c e o seu rush0X.c, no qual 0X corresponde ao número do rush. Por exemplo, rush00.c.
- Estes três ficheiros serão compilados juntos.
- O ficheiro ft_putchar.c deve incluir a função ft_putchar.
- Exemplo do main.c:

```
int main()
{
    rush(5, 5);
    return (0);
}
```

- Deverá, portanto, escrever a função rush tendo como parâmetro duas variáveis do tipo inteiro nomeadas respectivamente x e y.
- A função rush deverá exibir na tela um retângulo de x caracteres de largura, e y caracteres de altura.
- A sua função nunca deverá quebrar ou ficar em um ciclo infinito.
- O seu main será modificado na defesa para poder mudar os parâmetros da chamada à função rush. Aqui está um exemplo do que será testado:



Capítulo IV Rush 00

 \bullet rush(5,3) deverá mostrar:

• rush(5, 1) deverá mostrar:

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

• rush(1, 1) deverá mostrar:

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

• rush(1, 5) deverá mostrar:

Piscina C Rush 00 \bullet rush(4, 4) deverá mostrar: \$>./a.out 9

Capítulo V Rush 01

• rush(5,3) exibirá:

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

• rush(5, 1) deverá mostrar:

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

• rush(1, 1) deverá mostrar:

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

• rush(1, 5) deverá mostrar:

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

 \bullet rush(4, 4) deverá mostrar:

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

* *

* *

\**/

$>
```

Capítulo VI Rush 02

 \bullet rush(5,3) deverá mostrar:

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

• rush(5, 1) deverá mostrar:

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

• rush(1, 1) deverá mostrar:

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

• rush(1, 5) deverá mostrar:

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

• rush(4, 4) deverá mostrar:

```
$>./a.out
ABBA
B B
B B
CBBC
$>
```

Capítulo VII Rush 03

 \bullet rush(5,3) deverá mostrar:

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

• rush(5, 1) deverá mostrar:

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

• rush(1, 1) deverá mostrar:

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

• rush(1, 5) deverá mostrar:

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

• rush(4, 4) deverá mostrar:

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

Capítulo VIII Rush 04

 \bullet rush(5,3) deverá mostrar:

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

• rush(5, 1) deverá mostrar:

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

• rush(1, 1) deverá mostrar:

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

• rush(1, 5) deverá mostrar:

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

• rush(4, 4) deverá mostrar:

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

Capítulo IX

Submissão e avaliação

Entrega o teu trabalho no teu repositório Git, como habitual. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em confirmar os nomes dos teus ficheiros para ter a certeza que estão corretos.

Como estes trabalhos não são avaliados por um programa, sinta-se livre de organizar os ficheiros como preferir, desde que entregue os ficheiros obrigatórios e que cumpra com os requisitos.



Apenas precisas de entregar os ficheiros pedidos no enunciado deste projeto. $\,$