



# Piscina C

## Rush 00

*Sumário: Este documento é o tema do módulo C 00 da Piscina C da 42.*

# Conteúdo

<b>I</b>	<b>Instruções</b>	<b>2</b>
<b>II</b>	<b>Preâmbulo</b>	<b>4</b>
<b>III</b>	<b>Tema Comum</b>	<b>5</b>
<b>IV</b>	<b>Rush 00</b>	<b>6</b>
<b>V</b>	<b>Rush 01</b>	<b>8</b>
<b>VI</b>	<b>Rush 02</b>	<b>9</b>
<b>VII</b>	<b>Rush 03</b>	<b>10</b>
<b>VIII</b>	<b>Rush 04</b>	<b>11</b>

# Capítulo I

## Instruções

- Cada membro do grupo pode inscrever o grupo na defesa.
- O grupo deve estar inscrito na defesa.
- Qualquer pedido de esclarecimento sobre um dos temas complicará o tema.
- Você deve seguir o procedimento de entrega para todos os seus exercícios.
- O tema pode mudar até uma hora antes da entrega.
- Os exercícios são ordenados precisamente do mais simples ao mais complexo. Em nenhum caso nós daremos atenção, nem levaremos em conta um exercício complexo se um exercício mais simples não foi feito perfeitamente.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza cc.
- Se o seu programa não compila, você terá 0.
- Os exercícios do rush devem ser feitos em grupos de 2, 3 ou 4.
- O número de rush imposto pelo seu grupo seguirá esta regra:  
ordem alfabética da primeira letra do login do líder da equipe (de 1 a 26) módulo 5.
- Você deverá, portanto, fazer o tema indicado com seus parceiros impostos e se apresentar para a defesa na hora marcada com todos os seus parceiros.
- Depois da defesa, o projeto deverá ser finalizado. As defesas servem para apresentar e explicar o seu trabalho nos mínimos detalhes.
- Cada membro do grupo deve estar perfeitamente consciente do trabalho realizado, cada um dos membros será questionado, e a nota do grupo será baseada nas piores explicações.

- Obviamente, você deve fazer de tudo para entrar em contato com os seus parceiros: telefone, carta, pombo correio, bola de cristal, etc. Nenhuma desculpa será aceita em relação a problemas no grupo.
- Se, depois de realmente tentar de tudo, um de seus parceiros ainda estiver incomunicável: faça o seu rush e arranjaremos uma solução na defesa. Mesmo se foi o líder do grupo: vocês todos possuem acesso ao repositório.
- Você pode, opcionalmente, fazer mais de um tema para obter um bônus.
- A Moulinette não tem mente aberta. Ela não vai procurar entender o código que não respeita a Norma. A Moulinette utiliza o programa **norminette** para verificar a norma dos seus arquivos. Entenda que é uma tolice entregar um código que não passe na **norminette**.



Você deve absolutamente fazer o tema obrigatório perfeitamente para apresentar os temas bônus.



A partir de hoje, a Norminette deve ser lançada com a sinalização `-R CheckForbiddenSourceHeader`. A Moulinette também a utilizará.

# Capítulo II

## Preâmbulo

Letra da abertura de Pinky e Cérebro:

Pink: Cérebro, o que você quer fazer esta noite?

Cérebro: A mesma coisa que fazemos todas as noites, Pinky, tentar conquistar o mundo!

O Pinky e o Cérebro  
O Cérebro e o Pinky  
Um é um gênio,  
O outro imbecil  
Não cansam de tentar,  
O mundo dominar  
É o Pinky, o Pinky e o Cérebro  
O Cérebro e o Pinky  
e o Pinky e o Cérebro!


No final da noite  
Começa um novo plano  
Eles vão pensar em tudo  
Até o dia clarear

O Pinky e o Cérebro  
O Cérebro e o Pinky  
Eles fazem tudo na escuridão  
Não é fácil de explicar  
Nem de compreender  
O Pinky, o Pinky e o Cérebro  
O Cérebro e o Pinky e o Cérebro e o Pinky  
O Cérebro e o Pinky  
e o Cérebro e o Pinky.

O Cérebro e o Pinky e o Cérebro e o Pinky nurffff!

# Capítulo III

## Tema Comum

	Exercício : 00
rush0X	
Pasta de entrega : <i>ex00/</i>	
Arquivos para entregar : <code>main.c</code> , <code>ft_putchar.c</code> , <code>rush0X.c</code>	
Funções autorizadas : <code>write</code>	

- Os arquivos que devem ser entregues são o `main.c`, um `ft_putchar.c` e o seu `rush0X.c`, no qual 0X corresponde ao número do rush. Por exemplo, `rush00.c`.
- Exemplo de `main.c`:

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

- Você deverá, portanto, escrever a função `rush` tendo como parâmetro duas variáveis de tipo inteiro nomeadas respectivamente `x` e `y`.
- Sua função `rush` deverá exibir na tela um retângulo de `x` caracteres de largura, e `y` caracteres de altura.
- O seu `main` será modificado na defesa para poder mudar os parâmetros da chamada à função `rush`. Por exemplo, o tipo de coisa que será testada:

```
int      main()
{
    rush(123, 42);
    return (0);
}
```

# Capítulo IV

## Rush 00

- `rush(5,3)` exhibirá isto:

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

- `rush(5, 1)` isto:

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

- `rush(1, 1)` ceci :

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

- `rush(1, 5)` ceci :

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

- `rush(4, 4)` ceci :

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



# Capítulo V

## Rush 01

- `rush(5,3)` exhibirá isto:

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

- `rush(5, 1)` ceci :

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

- `rush(1, 1)` ceci :

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

- `rush(1, 5)` ceci :

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

- `rush(4, 4)`, isto:

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

# Capítulo VI

## Rush 02

- `rush(5,3)` exhibirá isto:

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

- `rush(5, 1)`, isto:

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

- `rush(1, 1)` ceci :

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

- `rush(1, 5)`, isto:

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

- `rush(4, 4)`, isto:

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

# Capítulo VII

## Rush 03

- `rush(5,3)` exhibirá isto:

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

- `rush(5, 1)` ceci :

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

- `rush(1, 1)` ceci :

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

- `rush(1, 5)`, isto:

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

- `rush(4, 4)`, isto:

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

# Capítulo VIII

## Rush 04

- `rush(5,3)` exhibirá isto:

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

- `rush(5, 1)`, isto:

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

- `rush(1, 1)` ceci :

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

- `rush(1, 5)`, isto:

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

- `rush(4, 4)`, isto:

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