



# Piscina C

## Rush 01

*Sumário: Este documento é o tema do Rush 01 da Piscina C da 42.*

# Conteúdo

<b>I</b>	<b>Instruções</b>	<b>2</b>
<b>II</b>	<b>Preâmbulo</b>	<b>3</b>
<b>III</b>	<b>Tema</b>	<b>4</b>
<b>IV</b>	<b>Bonus</b>	<b>6</b>
<b>V</b>	<b>Anexo</b>	<b>7</b>

# Capítulo I

## Instruções

- O grupo será inscrito para correções automaticamente.
- Não cancele a correção: não vai ter outra.
- Qualquer pedido de esclarecimento sobre o tema irá complicá-lo.
- Você deverá seguir o procedimento de entrega para o tema.
- O tema pode mudar até uma hora antes da entrega.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza gcc.
- Se o seu programa não compila, você terá 0.
- 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.
- O projeto deverá estar finalizado no momento da defesa. 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.
- 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 incommunicável: faça o rush e arranjaremos uma solução na defesa. Mesmo se foi o líder do grupo: vocês todos possuem acesso ao repositório.
- O seu trabalho deve estar dentro da Norma: seja muito rigoroso.
- Bom trabalho a todos.

# Capítulo II

## Preâmbulo

Aqui está o que a Wikipedia tem a dizer sobre a preguiça-real:

A preguiça-real é conhecida por ser o animal mais lento do mundo. O adulto pesa, em média, entre 4 e 8 kg, e seu tamanho é comparado ao de um cachorro de comprimento com uma cauda de 1,4 a 3,3 cm. Ela tem o pescoço curto, 4 longos membros de mesmo comprimento que no final possuem duas garras curvadas na frente e tr atrás, chegando a até 7 cm de comprimento. Sua cabeça é curta e plana, com um nariz arrebitado, orelhas rudimentares e olhos grandes.


Elas passam 80% do tempo dormindo. A preguiça se locomove muito lentamente. Elas dormem, comem, se acasalam, dão à luz e criam seus filhotes sem nunca descer da sua árvore, na qual ficam penduradas nos galhos de cabeça para baixo. Elas raramente descem para trocar de árvore quando a comida acaba ou para defecar, uma vez por semana.

Elas são camufladas pelas árvores, seus corpos castanho-esverdeados dobrados sobre si mesmos parecem um ninho de cupins ou uma parte da árvore aos olhos dos predadores. Em caso de ataque, elas se defendem com dentes e garras, mesmo que tenham um temperamento pacífico. Na verdade, elas contam mais com sua capacidade de camuflar com o fundo. Elas ficam imóveis por várias horas, mas se coçam regularmente, às vezes denunciando sua presença.

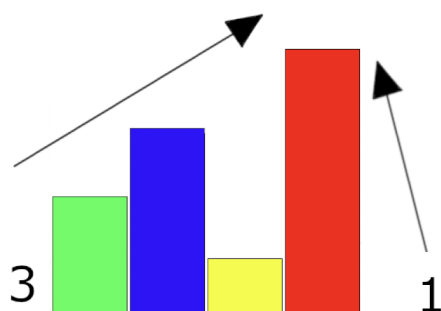
Diferente das preguiças, você não terá muito tempo para dormir esse final de semana. Que pena.

# Capítulo III

## Tema

	Exercício : 00
	rush-01
	Pasta de entrega : <i>ex00/</i>
	Arquivos para entregar : Todos os arquivos necessários
	Funções autorizadas : <code>write</code> , <code>malloc</code> , <code>free</code>

- O seu código inicial será compilado pelo comando `cc -Wall -Wextra -Werror -o rush-01 *.c`
- A sua pasta de entrega deverá conter tudo o que será necessário para a compilação do seu programa.
- Crie um programa que resolva o seguinte problema:
- Em um mapa 4 x 4, coloque caixas de tamanho 1 a 4 de modo que cada linha e coluna veja o número de caixas correto de todos os pontos de vista possíveis.
- Exemplo: a caixa de tamanho 3 esconde a caixa de tamanho 1, o que significa que existem apenas 3 caixas visíveis da esquerda. Da direita, a caixa de tamanho 4 esconde todas as outras caixas, portanto tem somente uma caixa visível.



- Cada uma das vistas (duas por linha e duas por coluna) terá um valor dado, de 1 a 4. O seu programa deve colocar as caixas corretamente, prestando atenção para ter apenas uma caixa de cada altura em cada linha e coluna.
- Se existirem muitas soluções, você deve exibir a primeira que encontrar.

- O programa será iniciado da seguinte maneira:

```
> ./rush-01 "col1up col2up col3up col4up col1down col2down col3down col4down row1left row2left  
row3left row4left row1right row2right row3right row4right"
```

- (cf. anexo 1)
- collup corresponde ao valor para ponto de vista da parte superior da coluna da esquerda. Cada valor deve estar entre 1 e 4.
- Você deverá exibir na saída a resolução da seguinte maneira:

```
> ./rush-01 "4 3 2 1 1 2 2 2 4 3 2 1 1 2 2 2" | cat -e  
1 2 3 4$  
2 3 4 1$  
3 4 1 2$  
4 1 2 3$
```

- (cf. anexos 2 e 3)
- Em caso de erro você deve escrever somente "Error" seguido de uma quebra de linha.

# Capítulo IV

## Bonus

Como bônus, você pode fazer um programa que resolva tabuleiros de tamanhos diferentes, como 3x3, 5x5, etc.

# Capítulo V

## Anexo

A seguir, você verá uma representação artística do seu programa. Você deve, obviamente, respeitar as instruções de entrega descritas na seção anterior. Essas visualizações têm o único propósito de te ajudar a entender o tema.

- Anexo 1:

	col1up	col2up	col3up	col4up	
row1left					row1right
row2left					row2right
row3left					row3right
row4left					row4right
	col1down	col2down	col3down	col4down	

- Representação da execução do programa com col\_up, col\_down, row\_left e row\_right
- Anexo 2:

	4	3	2	1	
4					1
3					2
2					2
1					2
	1	2	2	2	



- Substituindo os col\* e row\*, obtemos isto.
- Anexo 3:

	4	3	2	1	
4	1	2	3	4	1
3	2	3	4	1	2
2	3	4	1	2	2
1	4	1	2	3	2
	1	2	2	2	

- Seu programa deve preencher as caixas internas e retornar a resposta conforme solicitado no tema.