



# Piscina C

## Rush 01

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

*Versão: 7.2*

# Conteúdo

<b>I</b>	<b>Instruções</b>	<b>2</b>
<b>II</b>	<b>Preâmbulo</b>	<b>4</b>
<b>III</b>	<b>Enunciado</b>	<b>5</b>
<b>IV</b>	<b>Anexo</b>	<b>7</b>
<b>V</b>	<b>Submissão e avaliação</b>	<b>9</b>

# Capítulo I

## Instruções

- O grupo será automaticamente inscrito para defesa.
- No caso de ser cancelada, não terão direito a uma outra.
- Qualquer 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.
- A Moulinette compila com as *flags* -Wall -Wextra -Werror, e utiliza cc.
- Se o programa não compila, o grupo terá 0 como nota final.
- Deverá, portanto, executar o enunciado indicado com 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.
- 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.
- 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 tentar realmente tudo, 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.

# 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.

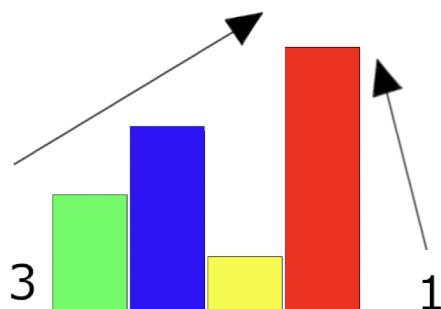
Diferente das preguiças, você não terá muito tempo para dormir esse final de semana. Que pena.

# Capítulo III

## Enunciado

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

- O seu código será compilado pelo comando `cc -Wall -Wextra -Werror -o rush-01 *.c`
- A sua pasta de entrega deverá conter tudo o que seja necessário para a compilação do programa.
- Crie um programa que resolva o seguinte problema:
- Num mapa 4 x 4, coloque caixas de tamanho 1 a 4 de modo que cada linha e cada coluna veja o número de caixas correto de todos os pontos de vista possíveis (esquerda/direita para as linhas; cima/baixo para as colunas)
- 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 dado valor. 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, 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)
- `col1up` corresponde ao valor para o ponto de vista da parte superior da coluna da esquerda. Cada valor deve estar entre 1 e 4.
- Este é o ÚNICO input aceitável para o seu programa. Qualquer outro input deve ser considerado um erro.
- Deverá exibir um *output* da seguinte forma:

```
> ./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, deve escrever apenas **Error** seguido de uma quebra de linha.
- No caso de querer fazer bonus, pode lidar com outros tamanhos de mapa (até 9x9!).
- Como habitual, caso o bonus funcione mas existam falhas na parte obrigatória, o resultado final é 0.

# Capítulo IV

## Anexo

Aqui fica uma representação artística do programa. Deve, obviamente, respeitar as instruções de entrega descritas na secção anterior.  
Essas visualizações têm o único propósito de ajudar a entender o enunciado.

- 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	

- O programa deve preencher as caixas internas e retornar a resposta conforme solicitado no enunciado.

# Capítulo V

## Submissão e avaliação

Entregue o seu trabalho no seu repositório `Git`, como habitual. Apenas o trabalho dentro do repositório será avaliado durante a defesa. Não hesite em confirmar os nomes dos 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 precisa de entregar os ficheiros pedidos no enunciado deste projeto.