

1 Paradigmas, I/O, Operações, Condicionais, Repetições e Sub-rotinas

1. **Divisor:** Dados dois inteiros positivos x e y , verifique se x é divisor de y . O retorno deve ser booleano (verdadeiro ou falso).
2. **Divisores:** Dado um número k , imprima todos os divisores de k .
3. **Máximo Divisor Comum (MDC):** Dados dois números m e n , imprima o máximo divisor comum entre m e n .
4. **Primo:** Dado um número x , verifique se ele é primo.
5. **Primos:** Dado um número k , imprima todos os números primos até k .
6. **Dama:** [Maratona de Programação 2008] O jogo de xadrez possui várias peças com movimentos curiosos. Uma delas é a dama, que pode se mover qualquer quantidade de casas na mesma linha, na mesma coluna, ou em uma das duas diagonais, conforme exemplifica a Figura 1.

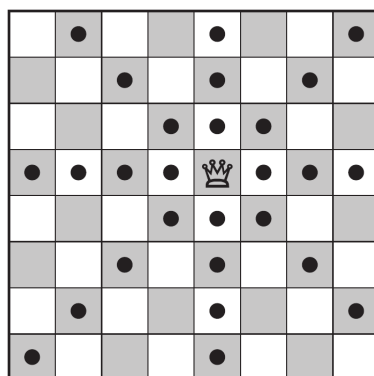


Figura 1: Movimentos possíveis da dama em um tabuleiro de xadrez.

Dada duas posições (x, y) e (m, n) em um tabuleiro de xadrez vazio (ou seja, um tabuleiro 8×8 , com 64 casas), calcule e imprima a quantidade mínima de movimentos que a dama precisa fazer para ir da posição (x, y) para a posição (m, n) . Exemplos:

<i>Entrada</i>		<i>Saída</i>
(x, y)	(m, n)	
(4, 4)	(6, 2)	1
(3, 5)	(3, 5)	0
(5, 5)	(4, 3)	2

7. **Acerola:** [Maratona de Programação 2008] Natural das Antilhas, a acerola (*Malpighia glabra* Linn, também conhecida como cereja das Antilhas) já era apreciada pelos nativos das Américas há muitos séculos. Mas o grande interesse por essa fruta surgiu na década de 1940, quando cientistas porto-riquenhos descobriram que a acerola contém grande quantidade de ácido ascórbico (vitamina C). A acerola apresenta, em uma mesma quantidade de polpa, até 100 vezes mais vitamina C do que a laranja e o limão, 20 vezes mais do que a goiaba e 10 vezes mais do que o caju e a amora.

Um grupo de amigos está visitando o Sítio do Picapau Amarelo, renomado produtor de acerola. Com a permissão de Dona Benta, dona do sítio, colheram uma boa quantidade de frutas, e pretendem agora fazer suco de acerola, que será dividido igualmente entre os amigos durante o lanche da tarde.

Conhecendo o número de amigos, a quantidade de frutas colhidas, e sabendo que cada unidade da fruta é suficiente para produzir 50 ml de suco, escreva uma função que receba como parâmetros o número N de amigos e a quantidade F de frutas colhidas, e imprima com precisão de duas casas decimais qual o volume, em litros, que cada amigo poderá tomar. Exemplos:

<i>Entrada</i>		<i>Saída</i>
N	F	
1	1	0.05
5	431	4.31
101	330	0.16

8. **Alarme Despertador:** [Maratona de Programação 2009] Daniela é enfermeira em um grande hospital e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores. Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar.

No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso. Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever uma função que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir. Exemplos:

<i>Entrada</i>				<i>Saída</i>
<i>Hora atual</i>	<i>Minuto atual</i>	<i>Hora alarme</i>	<i>Minuto alarme</i>	
1	5	3	5	120
23	59	0	34	35
21	33	21	10	1417

2 Listas

1. **Torneio:** Na primeira fase de um certo torneio de futsal, todos os times jogam entre si exatamente uma vez. Crie uma função que receba uma lista com o nome dos times e imprima a listagem dos jogos (um jogo por linha).
2. **Troca de Cartas:** [Maratona de Programação 2009] Alice e Beatriz colecionam cartas de Pokémon. As cartas são produzidas para um jogo que reproduz a batalha introduzida em um dos mais bem sucedidos jogos de videogame da história, mas Alice e Beatriz são muito pequenas para jogar, e estão interessadas apenas nas cartas propriamente ditas. Para facilitar, vamos considerar que cada carta possui um identificador único, que é um número inteiro.

Cada uma das duas meninas possui um conjunto de cartas e, como a maioria das garotas de sua idade, gostam de trocar entre si as cartas que têm. Elas obviamente não têm interesse em trocar cartas idênticas, que ambas possuem, e não querem receber cartas repetidas na troca. Além disso, as cartas serão trocadas em uma única operação de troca: Alice dá para Beatriz um sub-conjunto com N cartas distintas e recebe de volta um outro sub-conjunto com N cartas distintas.

As meninas querem saber qual é o número máximo de cartas que podem ser trocadas. Por exemplo, se Alice tem o conjunto de cartas $[1, 1, 2, 3, 5, 7, 8, 8, 9, 15]$ e Beatriz o conjunto $[2, 2, 2, 3, 4, 6, 10, 11, 11]$, elas podem trocar entre si no máximo quatro cartas. Escreva uma função que receba como parâmetros a lista de cartas que Alice tem e a lista de cartas que Beatriz possui, e imprima o número máximo de cartas que podem ser trocadas. As cartas de Alice e Beatriz são apresentadas em ordem não decrescente. Exemplos:

<i>Entrada</i>		<i>Saída</i>
<i>Cartas de Alice</i>	<i>Cartas de Beatriz</i>	
$[1000]$	$[1000]$	0
$[1, 3, 5]$	$[2, 4, 6, 8]$	3
$[1, 1, 2, 3, 5, 7, 8, 8, 9, 15]$	$[2, 2, 2, 3, 4, 6, 10, 11, 11]$	4

3. **Esquerda, Volver!:** [Maratona de Programação 2006] Este ano o sargento está tendo mais trabalho do que de costume para treinar os recrutas. Um deles é muito

atrapalhado, e de vez em quando faz tudo errado – por exemplo, ao invés de virar à direita quando comandado, vira à esquerda, causando grande confusão no batalhão. O sargento tem fama de durão e não vai deixar o recruta em paz enquanto este não aprender a executar corretamente os comandos.

No sábado à tarde, enquanto todos os outros recrutas estão de folga, ele obrigou o recruta a fazer um treinamento extra. Com o recruta marchando parado no mesmo lugar, o sargento emitiu uma série de comandos “esquerda volver!” e “direita volver!”. A cada comando, o recruta deve girar sobre o mesmo ponto e dar um quarto de volta na direção correspondente ao comando. Por exemplo, se o recruta está inicialmente com o rosto voltado para a direção norte, após um comando de “esquerda volver!” ele deve ficar com o rosto voltado para a direção oeste. Se o recruta está inicialmente com o rosto voltado para o leste, após um comando “direita, volver!” ele deve ter o rosto voltado para o sul. No entanto, durante o treinamento, em que o recruta tinha inicialmente o rosto voltado para o norte, o sargento emitiu uma série tão extensa de comandos, e tão rapidamente, que até ele ficou confuso, e não sabe mais para qual direção o recruta deve ter seu rosto voltado após executar todos os comandos. Você pode ajudar o sargento?

A função recebe como parâmetro uma lista com os comandos emitidos pelo sargento. Cada comando é representado por uma letra: ‘E’ (para “esquerda, volver!”) e ‘D’ (para “direita, volver!”). A função deve imprimir uma única linha da saída, indicando a direção para a qual o recruta deve ter sua face voltada após executar a série de comandos, considerando que no início o recruta tem a face voltada para o norte. A linha deve conter uma letra entre ‘N’, ‘L’, ‘S’ e ‘O’, representando respectivamente as direções norte, leste, sul e oeste. Exemplos:

<i>Entrada</i>	<i>Saída</i>
<i>Comandos</i>	
[‘D’, ‘D’, ‘E’]	L
[‘E’, ‘E’]	S

4. **Triangular inferior da transposta de uma matriz:** Defina um procedimento que receba uma matriz quadrada M como parâmetro. A função deve transformar a matriz M na matriz transposta de M , em seguida transformar essa matriz transposta em uma matriz triangular inferior e, por fim, imprimir a matriz resultante. Observações: (1) A função não pode criar listas auxiliares. (2) Uma matriz é triangular inferior quando todos os seus elementos acima da diagonal principal são iguais a 0.