

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 5

6 de Outubro de 2016

1. Considere o código em Python abaixo (assuma que no lugar de D seja usado o último dígito de seu RA):

```
j = 1

def main():
    a = 9 # Use o último dígito de seu RA.

    if a % 2 == 0:
        a = 2
    else:
        a = 3

    print(fun1(2,4))

    for i in range(1, 3):
        for j in range(1, 3):
            print(fun1(a, i+j))

def fun1(a, b):
    p = 1
    for i in range (1, b + 1):
        p = p * a
    return p + j

main()
```

- (a) Determine quais são as variáveis locais e globais deste programa. Para cada variável local identifique a que função esta pertence.
 - (b) Mostre o que será impresso na tela do computador quando for executado este programa (lembre-se de usar o último dígito de seu RA no lugar de D).
2. Escreva uma função **amigos(a, b)** que recebe dois números inteiros positivos *a* e *b* por parâmetro e determina se eles são amigos ou não, devolvendo 1 caso sejam amigos, e 0 caso contrário. Dois números são amigos se cada número é igual à soma dos divisores próprios do outro (os divisores próprios de um número *m* são os divisores estritamente menores que *m*). Por exemplo, os divisores próprios de 220 são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, cuja soma é 284; e os divisores próprios de 284 são 1, 2, 4, 71 e 142, cuja soma é 220. Logo, 220 e 284 são números amigos.
 3. Escreva uma função **mdc(m, n)** que calcule o máximo divisor comum de dois números *m* e *n* passados por parâmetro. Você deve utilizar a seguinte regra do cálculo do mdc onde

$$m \geq n$$

$$\text{mdc}(m, n) = m \text{ se } n = 0$$

$$\text{mdc}(m, n) = \text{mdc}(n, m \% n) \text{ se } n > 0$$

4. Escreva uma função `menor_base_log(n)` que recebe um valor inteiro positivo n como parâmetro e devolve um valor inteiro b tal que $b^k = n$ para algum inteiro k , e b seja o menor possível. Por exemplo, se $n = 27$ então o valor devolvido deve ser 3, já se $n = 12$ então o valor devolvido deve ser 12. Não use funções de bibliotecas na sua solução.
5. Um inteiro positivo n é **pitagórico** se existem inteiros positivos a e b tais que $a^2 + b^2 = n$. Por exemplo, 13 é pitagórico pois $2^2 + 3^2 = 13$.
 - (a) Escreva uma função `teste(a, b, n)` que recebe como parâmetro três inteiros a , b e n , e que devolve 1 caso $a^2 + b^2 = n$ e devolve 0 caso contrário.
 - (b) Utilize a função do item anterior e escreva uma outra função `pitagorico(n)` que recebe como parâmetro um inteiro positivo n e verifica se n é pitagórico, devolvendo 1 caso n seja pitagórico e 0 caso contrário.
6. Escreva uma função `media(v, tam)` que recebe um vetor de números reais e o seu tamanho por parâmetro e devolve a média aritmética dos números do vetor.
7. Escreva uma função `desvioPadrao(v, tam)` que recebe um vetor de números reais e o seu tamanho por parâmetro e devolve o desvio padrão dos números do vetor usando a seguinte fórmula:

$$\sqrt{\frac{1}{n-1} \left(\sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 \right)}$$

onde n é o número de elementos.

8. Escreva uma função `transposta(mat1, matRes, n)` que recebe como parâmetro duas matrizes quadradas de tamanho no máximo 30×30 , onde a primeira matriz corresponde a uma matriz de entrada e a segunda corresponde a uma matriz resposta. A função deve calcular a transposta da matriz de entrada e guardar o resultado na matriz resposta. A função também recebe como parâmetro um inteiro n que indica as dimensões das matrizes.
9. Uma matriz quadrada de inteiros é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna, a soma dos elementos da diagonal principal e da diagonal secundária são todos iguais. A matriz abaixo é um exemplo de quadrado mágico:

$$\begin{array}{ccc} 3 & 4 & 8 \\ 10 & 5 & 0 \\ 2 & 6 & 7 \end{array}$$

Escreva uma função `magico(mat, n)` que recebe como parâmetro uma matriz quadrada de tamanho no máximo 30×30 , e suas dimensões n , e determina se ela é um quadrado mágico devolvendo 1 caso seja e 0 caso contrário.

10. Escreva uma função `multiplica(mat1, mat2, matRes, n)` que recebe como parâmetro três matrizes quadradas de tamanho no máximo 30×30 , onde a primeira e a segunda matriz correspondem a entrada e a terceira corresponde a uma matriz resposta. A função deve calcular a multiplicação da primeira pela segunda matriz e guardar o resultado na matriz resposta. A função também recebe como parâmetro um inteiro n que indica as dimensões das matrizes.
11. Suponha que uma matriz binária represente ligações entre cidades, e que, se uma posição (i, j) possui o valor 1, então há uma estrada da cidade i para a cidade j . Seja o seguinte exemplo de matriz:

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Neste caso, há caminhos disponíveis da cidade 0 para a 1 e 2, e da 2 para 0.

Para cada item abaixo escreva uma função `verifica(mat, n, resposta)` que recebe como parâmetro uma matriz quadrada indicando as estradas entre as cidades, um inteiro n correspondendo as dimensões da matriz e um vetor resposta (que terá tamanho n).

- A função deve determinar as cidades com entrada e sem saída, indicando isto no vetor resposta, tal que `resposta[i]` recebe 1 caso a cidade i satisfaça esta propriedade e 0 caso contrário.
 - A função deve determinar as cidades com saída mas sem entrada, indicando isto no vetor resposta, tal que `resposta[i]` recebe 1 caso a cidade i satisfaça esta propriedade e 0 caso contrário.
 - A função deve determinar as cidades isoladas, indicando isto no vetor resposta, tal que `resposta[i]` recebe 1 caso a cidade i satisfaça esta propriedade e 0 caso contrário.
12. No jogo Sudoku temos uma matriz 9×9 dividida em 9 quadrados de 3×3 preenchidos previamente com alguns números entre 1 e 9 (veja o exemplo à esquerda abaixo). Uma solução para uma instância do jogo consiste no preenchimento de todas as posições vazias com números entre 1 e 9 respeitando-se as seguintes regras:
- (a) Não pode haver números repetidos em um mesmo quadrado, ou seja, cada número entre 1 e 9 deve aparecer exatamente uma vez em cada quadrado.
 - (b) Não pode haver números repetidos em nenhuma linha da matriz.
 - (c) Não pode haver números repetidos em nenhuma coluna da matriz.

Escreva uma função `solucao(mat)` que recebe uma matriz 9×9 por parâmetro que representa uma proposta de solução para um sudoku, e testa se a matriz é ou não uma solução para um sudoku, devolvendo 1 em caso verdadeiro e 0 caso contrário.

Veja abaixo um exemplo (direita) de uma matriz solução para um sudoku.

	2	5		1		9		
8			2	3			6	
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1	9		7		6		

Sudoku não resolvido

4	2	6	5	7	1	3	9	8
8	5	7	2	9	3	1	4	6
1	3	9	4	6	8	2	7	5
9	7	1	3	8	5	6	2	4
5	4	3	7	2	6	8	1	9
6	8	2	1	4	9	7	5	3
7	9	4	6	3	2	5	8	1
2	6	5	8	1	4	9	3	7
3	1	8	9	5	7	4	6	2

Solução