

Computação I - Modelo de Prova

A prova é individual e sem consulta. Responda as questões na folha de respostas, a lápis ou a caneta. Se tiver qualquer dúvida consulte o professor. A prova é frente e verso!

Boas práticas de programação também são parte da avaliação. Lembre-se de usar nomes significativos para variáveis e funções, organizar seu código com funções, colocar comentários e indentar corretamente.

1. (2 pontos) O número π pode ser calculado por meio de uma série numérica, a série de Leibniz, como mostrado abaixo:

$$\frac{\pi}{4} = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots$$

- (a) Escreva uma função que calcule o valor de π por meio da série de Leibniz até o n -ésimo termo do somatório, o número de termos que serão calculados deve ser passado como parâmetro .
- (b) Sabendo que a biblioteca math possui uma boa aproximação para o valor de π (`math.pi`), faça uma função que descubra quantos termos da série de Leibniz devem ser calculados utilizando a função da questão anterior para que o erro absoluto entre o valor de π da biblioteca math e o valor de π da série de Leibniz seja inferior a um erro passado como parâmetro para a função.

Dica: `math.fabs(numero)` retorna o valor absoluto (módulo) de um número.

2. (2 pontos) Seja a seguinte função:

```
# retorna uma lista com n valores aleatorios
# entre 0 e 100.
# int-> int
def sorteia_valores(n):
    valores = []
    for i in range(n):
        v = random.randint(0, 100)
        list.append(valores, v)
    return valores
```

- (a) Escreva uma função que retorna uma lista contendo 10 valores aleatórios entre 0 e 100, de forma que a diferença entre o valor máximo e o valor mínimo seja inferior a 50, usando obrigatoriamente a função `sorteia_valores`.
- (b) Reescreva a função `sorteia_valores` de forma que ela retorne uma lista de valores aleatórios entre 0 e 100 cuja soma seja no mínimo n , e que não tenha valores repetidos (não importa o número de elementos desta lista).
3. (2 pontos) Para cada uma das linhas da tabela, escreva os valores de ? de acordo com a chamada da função abaixo:

	Chamada de Função	Resultado
A	<code>crz_measures(["1;?", "1;3", "2;4"])</code>	{1: 1, 2: 16}
B	<code>crz_measures(["1;2", "?;3", "2;4", "2;5"])</code>	{1: 1, 2: 11}
C	<code>crz_measures(["4;8", "1;3", "2;4", "2;6"])</code>	?

```

# os comentários sobre esta função
# foram omitidos
def crz_measures(measures):
    dict = {}
    for data in measures:
        lista = str.split(data, ";")
        id_str, value_str = lista[0], lista[1]
        id = int(id_str)
        value = int(value_str)
        if id in dict:
            if value % 2 == 0:
                dict[id] += value
            else:
                dict[id] -= value
        else:
            dict[id] = value ** 2
    return dict

```

4. (4 pontos) Escreva as seguintes funções:

- *exibe_matriz*: recebe uma matriz de tamanho arbitrário e a exibe na tela.
- *ler_matriz*: recebe valores m e n e retorna uma matriz $m \times n$, pedindo que o usuário informe cada elemento da matriz.
- *triangular_inferior*: recebe uma matriz qualquer e retorna True se a matriz é triangular inferior, caso contrário retorna False. Uma matriz é triangular inferior se ela é quadrada e todos os elementos acima da diagonal principal são nulos.

Crie um programa completo (criando função *main*) que peça ao usuário valores m e n , leia uma matriz $m \times n$, exiba a matriz, e se a matriz informada for triangular inferior exiba a mensagem “A matriz é triangular inferior”, caso contrário exiba a mensagem “A matriz não é triangular inferior”. É obrigatório utilizar as funções pedidas acima.

BOA SORTE !