



Engenharia de Sistemas e Computação

Lista 1 – Algoritmos computacionais

Professor: Giomar Sequeiros

Período: 2023 – I

Instruções:

- Criar um arquivo .py com o nome do aluno sem espaços nem caracteres especiais (exemplo **joao_da_silva.py**)
- Identificar cada questão usando separadores adequados. Por exemplo
----- Questão 1-----
- O **código** deve estar devidamente **comentado e documentado** indicando os tipos de entrada e saída.

Q1. Crie uma função que receba o peso (Kg.) e a altura (m) de uma pessoa e retorne uma string indicando o seu Índice de Massa Corporal (IMC). O IMC é igual a **peso/altura²** e é classificado conforme a tabela abaixo:

IMC	Classificação
< 18,5	Abaixo do peso ideal
18,5 – 25	Peso normal
25 – 30	Excesso de peso
30 – 35	Obesidade (grau I)
35 – 40	Obesidade (grau II)
>40	Obesidade (grau III)

Q2. Escreva uma função que receba dois parâmetros: uma tupla com dois números e uma string. De acordo com a string recebida, e com a Tabela, a função deve realizar diferentes operações com os números e retornar o resultado. Obs. No caso de a string de entrada ser diferente das strings da Tabela, a função deve retornar uma string com a seguinte mensagem: 'A operação informada não foi reconhecida'.

String	descrição
'SOMA'	retorna a soma dos dois números
'MULT'	retorna a multiplicação dos dois números
'DIV'	retorna o número real resultado da divisão do primeiro número pelo segundo
'SUB'	retorna a o primeiro número subtraído do segundo
'MOD'	retorna o módulo (resto da divisão) do primeiro número pelo segundo
'POT'	retorna o primeiro número elevado ao segundo

Q3. Faça uma função que, dada uma frase, troque todas as vogais das palavras consideradas por i.

Exemplo:

- Frase lida "Levei meu cachorro para passear"
- Frase alterada "Livii mii cichirri piri pissiir"

Q4. Atualmente no Brasil, os números de telefone fixo têm 8 dígitos, enquanto os números de telefone celular têm 9 dígitos. Esses números são regionais, então, para completar a identificação do usuário no Brasil com base em um número, também é importante considerar o código de área (DDD), composto por 2 dígitos. É comum as pessoas fornecerem seu número telefônico tanto incluindo o código de área quanto não incluindo.

Seguindo apenas essas regras de tamanho, alguns exemplos de números de telefone válidos no Brasil são:

21912316165
1132316165
71912316165
323231616
32316165
912316165

Faça uma função que receba como entrada uma string contendo uma sequência de dígitos, que supostamente corresponde ao número de telefone informado por um usuário (parta do princípio de que essa string só contém dígitos). Sua função deve identificar se o número de telefone é válido no Brasil, ou seja, se ele se encaixa em um dos padrões aceitáveis. Caso seja válido, sua função deve retornar uma tupla com duas strings: a primeira contendo exatamente dois dígitos correspondentes ao DDD (ou uma string vazia, caso o DDD não tenha sido informado), e a segunda contendo 8 ou 9 dígitos correspondentes ao número de telefone sem o DDD. Caso o número seja inválido, sua função deve retornar uma tupla com duas strings vazias

Q5. Faça uma função em Python, que receba uma tupla contendo 4 informações: o nome do aluno e três notas. Sua função deve retornar uma tupla, cujo primeiro elemento é o nome do aluno, o segundo elemento é a média e o terceiro elemento é a situação do aluno, representada por uma string. Se a média das três notas do aluno for maior ou igual a 7 (inclusive), a função deverá retornar: (<nome>, <media>, 'aprovado, Parabéns!'). Se a média do aluno for menor que 7, porém maior ou igual a 5, a função deve retornar: (<nome>, <media>, 'aprovado'). Se a média for menor que 5, a função deve retornar: (<nome>, <media>, 'reprovado'). A média deve ser retornada com uma casa decimal apenas.

Q6. Questão OBI (Olimpiada Brasileira de Informática - 2007, Fase 1, Nível 1) - (Detectando Colisões)

Deteção de colisão é uma das operações mais comuns (e importantes) em jogos eletrônicos. O objetivo, basicamente, é verificar se dois objetos quaisquer colidiram, ou seja, se a interseção entre eles não é vazia. Isso pode ser usado para saber se duas naves colidiram, se um monstro bateu em uma parede, se um personagem pegou um item, etc. Para facilitar as coisas, muitas vezes os objetos são aproximados por figuras geométricas simples (esferas, paralelepípedos, triângulos etc). Neste problema, os objetos são aproximados por retângulos em um plano 2D.

Escreva uma função chamada `colisao` que, dados dois retângulos, determine se eles se interceptam ou não. Cada retângulo é determinado pelas coordenadas x e y de dois de seus vértices diametralmente opostos, representando a diagonal que vai da esquerda para a direita e de baixo para cima. Os lados de cada retângulo são sempre paralelos aos eixos x e y .

Entrada: Os parâmetros de entrada são duas tuplas com quatro valores inteiros cada uma, representando as coordenadas do primeiro retângulo e as coordenadas do segundo retângulo.

Saída: Sua função deve retornar o valor booleano `True` caso haja interseção ou `False` caso não haja.

Exemplos:

Entrada: (0,0,1,1), (0,0,1,1) ; Saída: `True`

Entrada: (0,0,2,2), (1,1,3,3) ; Saída: `True`

Entrada: (0,0,1,1), (2,2,3,3) ; Saída: `False`

Q7. Faça uma função chamada `maiores_que`, que recebe uma lista de números inteiros e um número inteiro `n`. A função deve retornar outra lista contendo todos os números da lista original que são maiores que `n`, ordenados em ordem crescente.

Q8. Faça uma função chamada `repetidos` que receba como entrada uma lista de números e retorne o número de vezes que um elemento da lista é igual ao elemento anterior.

Exemplo: `repetidos([1, 4, 3, 3, 2, 3, 3, 3, 5, 4, 6, 6, 7, 6, 8, 8, 7])`

Resposta: 6

Q9. Escreva uma função que receba um número inteiro positivo como argumento e verifique se é um número perfeito. Um número perfeito é aquele que é igual à soma de seus divisores próprios (excluindo ele mesmo).

- Exemplos de números perfeitos

- O número 6 é um número perfeito, pois a soma de seus divisores próprios (1, 2, 3) é igual a ele mesmo: $1 + 2 + 3 = 6$.
- O número 28 é um número perfeito, pois a soma de seus divisores próprios (1, 2, 4, 7, 14) é igual a ele mesmo: $1 + 2 + 4 + 7 + 14 = 28$.

Q10. Uma pessoa **investe** uma **quantidade** (em reais) em uma conta de **poupança** que rende uma taxa de juros ao ano. Admitindo que todos os juros são deixados em depósito na conta, crie uma função que calcule e imprima a quantia na conta ao final de cada ano, ao longo de **n** anos. Use a seguinte fórmula para determinar estas quantias: $a = p(1 + r)^n$

Onde:

- **p** é a quantia investida originalmente (i.e., o valor principal)
- **r** é a taxa anual de **juros**.
- **n** é o número de **anos**

a é a quantia existente em depósito no final do n-ésimo ano

Exemplo de saída para R\$1000,00 em 10 anos e 5% (0.05) de juros ao ano

Ano Saldo na conta

1	1050.00
2	1102.50
3	1157.62
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89