

Computação I - Python

Laboratório 4

ATENÇÃO

A partir de agora, para cada um dos exercícios a seguir:

- coloque um comentário dizendo o que a função faz;
- coloque um comentário dizendo os tipos de dados dos parâmetros de entrada e dos resultados.

Exemplo:

int, int → float

no caso de uma função cuja entrada são dois inteiros e o resultado é um float;

- escolha nomes elucidativos para suas funções e parâmetros;
- Diga os casos de teste usados.

1. Escreva uma função que receba duas strings a e b , e retorne a concatenação delas no formato $abba$.
2. Dizem que para saber o seu número da sorte basta você seguir os seguintes passos:
 1. Multiplique sua idade por 4.
 2. Some 8.
 3. Multiplique o resultado por 60.
 4. Divida por 240.
 5. Some 22.
 6. Subtraia a sua idade.
 7. PARABÉNS! Esse é seu número da sorte.

Escreva um função que receba o nome e a idade de uma pessoa e retorne a frase: "*Parabéns fulano! seu número da sorte é x!*", onde *fulano* é o nome da pessoa, e x é o número da sorte encontrado.

3. Escreva uma função que receba duas strings de no mínimo 15 caracteres e retorne a concatenação da primeira, sem os cinco primeiros caracteres, com a segunda, sem os últimos dez caracteres.
4. Escreva uma função que receba uma string s , um caractere x e um número inteiro i entre 0 e o comprimento da string menos 1, e retorne uma string igual a s , exceto que o elemento da posição i deve ser substituído pelo caractere x .

5. Escreva uma função que receba uma string e retorne essa string no meio dela mesma. Por exemplo, ao receber a string "abcd", a função deve retornar "ababcdcd". Outro exemplo: se receber "abcde", a função deve retornar "ababcdecde"
6. Escreva uma função que receba uma string e insira o caractere "#" no início, no meio e no final dela. Por exemplo, se a entrada for "abcd", a saída deve ser "#ab#cd#". Outro exemplo: se receber "abcde", a função deve retornar "#ab#cde#".
7. Escreva uma função que receba uma string e a rotacione 3 posições para a esquerda. Por exemplo, se a entrada for "abcdefg", a função deve retornar "efgabcd". Assuma que a string passada tem no mínimo 3 caracteres.
8. Escreva uma função que receba uma string e um número inteiro x e rotacione a string x posições para a esquerda. Assuma que a string tem pelo menos x caracteres.
9. Escreva uma função similar à anterior, só que agora considere que a string passada pode ter qualquer tamanho, inclusive menor que x .
10. Escreva uma função que receba duas datas no formato "DD/MM/AAAA", sendo a segunda maior que a primeira, e calcule o total de dias passados entre uma data e outra. A frase retornada deve ser "O total de dias é x ", onde x é o total achado. Considere que todo mês tem 30 dias. E que o ano tem 365 dias. Exemplo: Se as datas são "02/03/1982" e "01/02/1983", o total de dias é 334.

11. Questão OBI (Olimpíada Brasileira de Informática - 2007, Fase 1, Nível 1) - (Detectando Colisões)
Detecção 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 é diferente de vazio. Isso pode ser usado para saber se duas naves colidiram, se um monstro bateu numa 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 num plano 2D.

Escreva uma função que, dados dois retângulos, determine se eles se interceptam ou não.

Entrada: Os parâmetros de entrada são 8 números inteiros representando as coordenadas do primeiro retângulo ($retanguloX_{1,0}, retanguloY_{1,0}, retanguloX_{1,1}, retanguloY_{1,1}$) e as coordenadas do segundo retângulo ($retanguloX_{2,0}, retanguloY_{2,0}, retanguloX_{2,1}, retanguloY_{2,1}$). Considere que os lados de cada um dos retângulos são sempre paralelos aos eixos x e y .

Saída: A 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*