Lista_02

September 3, 2020

1 Lista de Exercícios 2

*Alguns exercícios foram retirados do livro: The Python Workbook - Ben Stephenson.

1.1 QUESTÃO 1:

Crie uma função que recebe como parâmetro de entrada dois números inteiros positivos e encontre o seu **Máximo Divisor Comum** (MDC).

[ ]:

1.2 QUESTÃO 2:

Agora, crie uma função que recebe como parâmetro de entrada dois números inteiros positivos e encontre o seu **Mínimo Múltiplo Comum** (MMC).

[ ]:

1.3 QUESTÃO 3:

No iníco do curso, vimos como fazer conversão de números binários (base 2) para decimais (base 10). Portanto, crie uma função que recebe como parâmetro de entrada um número binário e calcula o número decimal correspondente.

[ ]:

1.4 QUESTÃO 4:

Agora, escreva uma função que faça o inverso, ou seja, que faça a conversão de um número decimal (base 10) em número binário (base 2).

[ ]:

1.5 QUESTÃO 5:

Crie uma função que começa selecionando um número inteiro aleatório entre 1 e 100. Armazene esse valor inicial como sendo o maior número inteiro encontrado até o momento. Em seguida, gere aleatoriamente mais 99 números inteiros, verificando sempre se o novo valor é maior do que o máximo encontrado anteriormente. Caso seja maior, a função contabiliza que ocorreu uma atualização

do valor máximo. Ao final, sua função deverá retornar o valor máximo encontrado e a quantidade de veses que o valor máximo foi atualizado durante o processo.

Por exemplo,

Sorteio	Valor	Máximo	Atualizações
1	10	10	
2	5	10	
3	15	15	1
4	14	15	
5	20	20	2

Em 5 sorteios, o valor máximo encontrado foi 20 após 2 atualizações.

[ ]:

1.6 QUESTÃO 6:

Crie uma função que encontre o número mínimo de vezes que você tem que jogar uma moeda antes de poder ter três lançamentos consecutivos que resultam no mesmo resultado (ou todos os três são CARA (C) ou todos os três são COROA (K))?

[ ]:

1.7 QUESTÃO 7:

Escreva uma função que determina se uma senha é válida ou não. Uma boa senha é defnida segundo os seguintes critérios:

- possui pelo menos 8 caracteres
- contém pelo menos uma letra maiúscula
- contém pelo menos uma letra minúscula
- contém pelo menos um número

Sua função deve retornar verdadeiro (True) se a senha informada for válida ou falso (False), caso contrário.

[ ]:

1.8 QUESTÃO 8:

Uma **data mágica** é uma data em que o dia multiplicado pelo mês é igual ao ano de dois dígitos. Por exemplo, 10 de junho de 1960 é uma data mágica porque junho é o sexto mês e 6 vezes 10 é 60, que é igual ao ano de dois dígitos.

Escreva uma função que determina se uma data é ou não uma data mágica.

[ ]:

1.9 QUESTÃO 9:

Dizemos que dois números são **números amigos** se cada um deles é igual a soma dos **divisores próprios** do outro. Os divisores próprios de um número positivo n são todos os divisores inteiros positivos de n exceto o próprio n.

Crie uma função que verifica se dois números são amigos (retorna True) ou não (retorna False).

[ ]:

1.10 QUESTÃO 10:

O CPF é formado por 11 dígitos numéricos que seguem a máscara "###.###.###-##", a **verificação do CPF** acontece utilizando os 9 primeiros dígitos e, com um cálculo simples, verificando se o resultado corresponde aos dois últimos dígitos (depois do sinal "-").

VALIDAÇÃO DO PRIMEIRO DÍGITO

- multiplicar os 9 primeiros dígitos pela sequência decrescente de números de 10 à 2 e somar os resultados;
- multiplicar esse resultado por 10 e dividir por 11;
- se o RESTO for igual ao primeiro dígito verificador (primeiro dígito depois do '-'), a primeira parte da validação está correta (se o resto da divisão for igual a 10, nós o consideramos como 0).

VALIDAÇÃO DO SEGUNDO DÍGITO

- multiplicar os 10 primeiros dígitos pela sequência decrescente de números de 11 à 2 e somar os resultados;
- multiplicar esse resultado por 10 e dividir por 11;
- se o RESTO for igual ao segundo dígito verificador (segundo dígito depois do '-'), a validação está correta!

Sua tarefa é criar uma função que verifica se um CPF é válido. O parâmetro de entrada dessa função é uma string da seguinte forma: "###.###.###-##" e esse formato deve ser verificado antes de efetuar os cálculos da validação.

Atente-se ao fato de que alguns CPFs passam nessa validação, mas ainda são inválidos. É o caso dos CPFs com dígitos repetidos ("111.111.111-11", "222.222.222-22", …). Adicione essa verficação!

[ ]: