

Lista de exercícios - 03

Exercícios de operações com números inteiros

Estes exercícios devem ser entregues no Google Classroom. Para cada um dos exercícios, crie um arquivo fonte Python com o respectivo nome de acordo com a seguinte regra: SUASINICIAIS-LP-XX-Ex-YY.py, onde XX e YY são o número da lista e o número do exercício, respectivamente. Por exemplo, se o professor resolvesse o exercício número 3, o nome do arquivo seria **SUASINICIAIS-LP-03-Ex-03.py**

Introdução

Os exercícios a seguir foram preparados para ajudá-lo a desenvolver suas habilidades na resolução de problemas utilizando operações com números inteiros. Além disso, os exercícios vão ajudá-lo a adquirir familiaridade com conceitos básicos e com a sintaxe da linguagem Python. Para completar os exercícios, você vai precisar usar alguns (ou todos) os recursos do Python abaixo:

- Escrever e testar código Python em algum ambiente de desenvolvimento (por exemplo, o Visual Studio Code)
- Gerar saída em texto utilizando a função print
- Ler entrada no teclado com a função input, inclusive fazendo conversão de tipo de dado quando necessário
- Construir expressões aritméticas com números inteiros (int) usando os operadores aritméticos inteiros do Python tais como // e %
- Controlar como a saída é exibida, por meio de especificações de formatação

Questões:

1. **Centena, dezena e unidade.** Dado um **número inteiro** de três algarismos $N = CDU$ (onde C é o algarismo das centenas, D é o algarismo das dezenas e U o algarismo das unidades), considere o número M constituído pelos algarismos de N em ordem inversa, isto é, $M = UDC$. Faça um programa Python para gerar M a partir de N (p.ex.: $N=123 \rightarrow M=321$).

DICA: para resolver este problema, seu programa deve receber do usuário **um único número inteiro** com 3 dígitos, e deve “extrair” os dígitos da centena, da dezena e da unidade por meio de operações de divisão inteira e resto de divisão. Depois, para gerar o número M, basta fazer a operação inversa. Por exemplo, para o carácter da unidade virar carácter da centena, basta multiplica-lo por 100.

2. **Decompondo uma data.** Admitindo que uma data é lida por um algoritmo em uma variável inteira, e não em uma variável do tipo data, crie um programa Python que leia uma data no **tipo de dados de número inteiro** no formato DDMMAA e imprima essa data no formato AAMMDD, onde:
 - a letra D corresponde a dois algarismos representando o dia;
 - a letra M corresponde a dois algarismos representando o mês;
 - a letra A corresponde aos dois últimos algarismos representando o ano."

DICA: este problema é parecido com o anterior. A data deve ser lida como um único número inteiro e você deve extrair os caracteres por meio de divisões inteiras e restos de divisão. Para simplificar, ignore datas que começam com 0 (por exemplo: mês 04, dia 09, etc)

3. **Número de matrícula.** Suponha que uma escola utilize, como código de matrícula, um número inteiro no formato AASDDD, onde:
- os dois primeiros dígitos, representados pela letra A, são os dois últimos algarismos do ano da matrícula;
 - o terceiro dígitos, representado pela letra S, vale 1 ou 2, conforme o aluno tenha se matriculado no 1º ou 2º semestre;
 - os três últimos dígitos, representados pela letra D, correspondem à ordem da matrícula do aluno, no semestre e no ano em questão.

Crie um programa Python que leia o número de matrícula de um aluno e imprima o ano e o semestre em que ele foi matriculado.

4. **Dígito verificador.** Dígito verificador é um recurso muito usado em números de documentos e contas bancárias para garantir que o número principal (da conta bancária, por exemplo) esteja correto. A ideia consiste em gerar um novo número (o dígito verificador) a partir dos próprios números da conta ou do documento. Assim, se o usuário cometer algum erro ao digitar os números, isso pode ser conferido pelo cálculo do dígito verificador.

Faça um programa Python que leia **um número inteiro** de cinco algarismos (variável **codigo**) e gere e imprima o dígito verificador (**digitoV**) módulo 7 para o mesmo.

O dígito verificador pode ser calculado da seguinte maneira: supondo que os cinco algarismos do código são **ABCDE**, uma forma de calcular o dígito desejado, com módulo 7 é:

$$S = 6*A + 5*B + 4*C + 3*D + 2*E$$

digito_verificador = resto da divisão de S por 7

Testes e exemplos de cada questão:

1. Exemplo, se N = 937, o programa deve imprimir o número 739
2. Exemplo: a data 141021 (14 de outubro de 2021), deve ser impressa como 211014
3. Um número de matrícula 212034 deve resultar ano 21 e semestre 2.
Opcional: você pode escrever o ano completo como 2021, e escrever o semestre por extenso ("Primeiro semestre" / "Segundo semestre", para isso, você precisa usar o comando **if**, que ainda não aprendemos nas aulas teóricas).
4. Exemplo: o código 84729 deve gerar um dígito verificador igual a 1