

1 Introdução

Resolva as seguintes questões em Python, **utilizando os recursos vistos até o momento na disciplina**. Esta avaliação é opcional, e não vale nota ou pontuação. Cada aluna(o) (ou grupo de alunos) poderá enviar ao professor seu material para recebê-lo comentado, mas tal envio é opcional.

1. Escreva um programa que lê um número inteiro X a partir do teclado e exibe a versão *espelhada* deste número, isto é, com os seus algarismos em ordem inversa. Assuma que o número tem exatamente três dígitos, todos diferentes de zero.

Por exemplo, ao ler 629, o programa deve exibir 926 na tela.

Como de costume, há mais de uma maneira de escrever um programa que realize essa tarefa. Uma das maneiras de fazê-lo é usar os operadores de divisão (inteira) e de resto `%` e tirando vantagem do fato de que $X \% 10$ nos dá o algarismo das unidades de X . Além disso, $X // 10$ elimina esse algarismo e nos deixa com o restante do número (do algarismo das dezenas em diante).

2. Escreva um programa que lê uma palavra de exatamente 5 letras a partir do teclado e imprime uma mensagem informando se a palavra é um palíndromo ou não. Se você não sabe ou não lembra o que é um palíndromo, consulte a Wikipedia.
3. Escreva um programa que lê um número inteiro positivo a partir do teclado (você pode confiar que a usuária vai digitar um número inteiro e positivo). Seu programa deve imprimir cinco frases na tela, informando se o número lido é divisível por 2, por 5, por 7, por 10, e por 35. Para cada um destes números, uma frase deve ser impressa, informando se o valor lido é, ou não, divisível por aquele outro número.

Após escrever sua primeira versão do programa, preocupe-se em evitar a realização de certos testes desnecessários. Por exemplo, ao saber que o número é divisível por 10 podemos inferir que ele é divisível por 2 e por 5, isto é, economizamos dois testes quando isso acontece: se é divisível por 10, então eu já posso imprimir as frases sobre divisibilidade por 2, por 5, e por 10. Isso tem a ver com o fluxo que seu programa segue quando é executado.

Talvez seu programa fique mais longo com esta modificação, mas perceba: o objetivo da questão **NÃO** é que seu programa tenha menos instruções do tipo `if` do que em sua versão original; o objetivo é que, para certos valores fornecidos pela usuária, seu programa realize menos instruções `if` do que ele realizaria em sua versão original. No pior caso, uma execução de seu programa deve realizar cinco testes, nunca mais do que isso. Mas, se o número digitado for, por exemplo, 70, são suficientes apenas dois testes. Existem vários outros casos similares, além do 70. Portanto, não crie um teste específico para o valor 70.

2 Entrega

Via SIGAA, por gentileza.