

## Computação I

Lista de exercícios 2 – Estrutura Condicional e Expressões Booleanas

**Atenção! Leia as instruções antes de fazer a lista! A padronização do nome do arquivo e dos nomes das funções é muito importante e está explicada no arquivo de instruções!**

Data de entrega: 11/05/2022

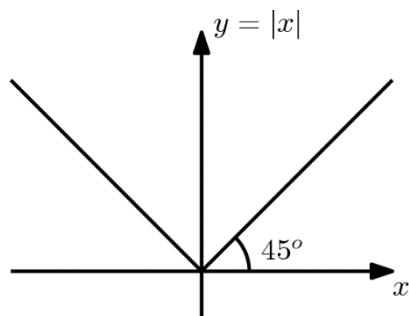
Não utilize nenhum módulo, método ou função já existente no Python exceto pela função print e pela função type (de forma simplificada, função é tudo o que precisa ser chamado com parênteses, o input, por exemplo é uma função, não use input; e método é tudo o que precisa ser chamado com parênteses, mas que está associado a uma variável). Não utilize laços de repetição ou recursão nessa lista. Não é necessário testar se os dados passados por argumento são válidos a menos que pedido na questão. Testar o código é uma boa prática de programação! Veja o arquivo de testes para exemplos de entradas e saídas.

1. Escreva uma função em Python que recebe um número por argumento e retorna 0 se esse número for zero, 1 se esse número for diferente de zero e ímpar e 2 se o número for diferente de zero e par.
2. Escreva uma função em Python que recebe um argumento x e retorna um valor y dada pela seguinte função por partes:

$$y = \begin{cases} 0, & \text{se } x < 0 \\ 3, & \text{se } 0 \leq x < 1 \\ 3 - 2(x - 1), & \text{se } 1 \leq x < 3 \\ (x - 3)^2 + 4(x - 3), & \text{se } 3 \leq x < 4.5 \\ -(x - 4.5)^3 + 5x^2 - 6(x - 5) + 2, & \text{se } x \geq 4.5 \end{cases}$$

Para essa questão, **utilize uma única estrutura condicional completa, com if, elifs, e else.**

3. Escreva uma função em Python que receba 3 números inteiros ou floats e retorna 1 caso o valor da diferença entre o primeiro argumento e o segundo argumento seja, em módulo, menor do que o terceiro argumento, e 0 caso contrário. A função módulo  $y = |x|$  é dada por:



**Atenção! Não é permitido o uso da função “abs()” do Python nessa questão (vide instruções)!**

4. Considere um jogo no qual um jogador tem a tarefa de encontrar o maior número de bolas vermelhas dentro de um cesto com bolas de várias cores no menor tempo possível. O jogador pode finalizar a tarefa a qualquer momento, mesmo que nem todas as bolas tenham sido encontradas. A pontuação base do jogador é dada pela seguinte regra:
- Se o jogador encontrar todas as bolas vermelhas disponíveis, então ele ganha 100 pontos.

- Se o jogador não encontrar todas as bolas vermelhas disponíveis, mas achar mais do que 70% desse total, então ele ganha 50 pontos.
- Se o jogador encontrar entre 30% (inclusive) e 70% (inclusive) do total de bolas vermelhas disponíveis, então ele ganha 25 pontos.
- Se o jogador encontrar menos de 30% das bolas vermelhas disponíveis, então ele não pontua.

A pontuação final do jogador é definida segundo o tempo que ele levou para concluir a tarefa, e segue a seguinte regra:

- Se o jogador concluiu a tarefa até antes do tempo limite estipulado (inclusive), então a pontuação final é igual à pontuação base.
- Se o jogador concluiu a tarefa até 10 minutos depois do tempo limite estipulado, então a pontuação final é 80% da pontuação base obtida.
- Se o jogador concluiu a tarefa entre 10 (inclusive) e 30 (inclusive) minutos depois do tempo limite estipulado, então a pontuação final é 40% da pontuação base obtida.
- Se o jogador concluiu a tarefa mais de 30 minutos depois do tempo limite estipulado, então o jogador não pontua.

Escreva uma função em Python que calcule a pontuação final de um jogador neste jogo. A função recebe 4 argumentos inteiros, nesta ordem: a quantidade de bolas vermelhas encontradas pelo jogador, a quantidade total de bolas vermelhas totais estavam no cesto, o tempo limite da tarefa (em minutos), e o tempo em que o jogador concluiu a tarefa (em minutos). A função deve retornar um número que representa a pontuação final do jogador.

5. Escreva uma função em Python que recebe, por argumento, e nessa ordem, a quantidade total de aulas e o número de faltas de um aluno. A função deve calcular a porcentagem de faltas desse aluno em relação ao número total de aulas. O aluno tem direito a faltar um número de aulas correspondente a até 30% da quantidade total de aulas. Por exemplo: se o total de aulas for 10, o aluno pode faltar a, no máximo, 3 aulas. Assuma que o número máximo permitido de faltas sempre será um número inteiro. Se a porcentagem for maior que 30%, a função deve retornar 0, o número de faltas excedidas e a porcentagem em si. Se a porcentagem for menor ou igual a 30%, a função deve retornar 1, a quantidade de vezes que o aluno ainda pode faltar e a própria porcentagem. Antes de calcular a porcentagem, no entanto, verifique se os valores recebidos são válidos: os números devem ser inteiros maiores ou iguais a 0 e o número de faltas deve ser menor ou igual à quantidade total de aulas. Utilize os operadores booleanos. Caso os valores recebidos não sejam válidos, retorne -1.
6. O jogo de par ou ímpar é um jogo simples de sorte jogado por duas pessoas. Em um determinado instante de tempo cada jogador deve mostrar uma quantidade de dedos em uma mão, onde essa quantidade é definida pelo próprio jogador e só pode ser vista pelo adversário no instante de tempo definido. Se o somatório total dos dedos mostrados for par, um dos jogadores é o vencedor. Se for ímpar, o outro é o vencedor. Faça uma função que define a quantidade de vitórias de cada jogador em um jogo de par ou ímpar com melhor de três. Na “melhor de três” o jogo de par ou ímpar é jogado três vezes e o vencedor é aquele que venceu em no mínimo duas das três vezes. Sua função deve receber por argumento seis inteiros entre 0 e 5, na seguinte ordem: as três jogadas do jogador 1 (referentes aos jogos 1, 2 e 3) e as três jogadas do jogador 2 (referentes aos 1, 2 e 3). A função deve retornar a quantidade de vitórias do jogador 1 e a quantidade de vitórias do jogador 2, sabendo que o jogador 1 escolheu o par e que o jogador 2 escolheu o ímpar. Considere que o 0 conta como uma vitória do par.
7. Considere que você deseja cozinhar uma determinada quantidade de bolos, baseada na quantidade de ovos e farinha que você tem à disposição. Escreva uma função em Python que receba 4 argumentos, nesta ordem: a quantidade disponível de ovos, a quantidade necessária de ovos para fazer um bolo, a quantidade

disponível de farinha (em gramas) e a quantidade necessária de farinha (em gramas) para fazer um bolo. O programa deve retornar 3 valores, nesta ordem: a quantidade de bolos que é possível fazer considerando todos os ingredientes disponíveis, a quantidade de caixas de ovos que seria necessário comprar para fazer um bolo extra, e a quantidade de farinha (em gramas) que seria necessário comprar para fazer um bolo extra. Considere que as caixas de ovos contêm sempre 12 ovos, e que os ovos são vendidos apenas em caixas. Por exemplo: assumindo que sobrou uma quantidade suficiente de farinha para uma nova porção, se sobraem 8 ovos, e 10 ovos forem necessários para fazer mais um bolo, então a quantidade de caixas a serem compradas será 1, apesar de apenas 2 ovos a mais serem necessários (não é possível comprar apenas 2 ovos, já que eles são vendidos em caixas contendo 12 ovos). A função deve verificar se os argumentos de entrada são válidos: as quantidades disponíveis de ovos e farinha devem ser inteiros maiores ou iguais a 0, enquanto que as quantidades necessárias de ovos e farinha devem ser inteiros positivos apenas. Caso algum argumento seja inválido, a função deve retornar (-1,-1,-1).