

Computação I

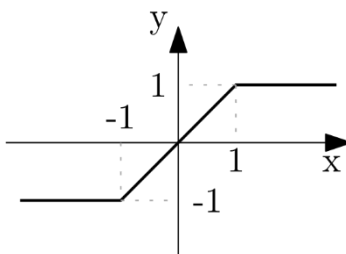
Lista de exercícios 2 – Estrutura Condicional e Expressões Booleanas

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: 28/09/2022

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 dois números não-nulos por argumento, e retorna 1 se ambos os números tiverem o mesmo sinal, e retorna 0 caso contrário.
2. Em ciência da computação, redes neurais são modelos computacionais usados na área de inteligência artificial. A saída dos neurônios é comumente descrita por uma função de ativação, que, muitas vezes, limita o valor de saída a um máximo e um mínimo, como a função mostrada no gráfico abaixo. Escreva uma função em Python que recebe um número x por argumento e retorna o y equivalente, onde y é dado pela seguinte função por partes:



3. Faça uma função em Python para realizar a conversão entre Celsius, Fahrenheit ou Kelvin. Essa função deve receber dois argumentos (respeite a ordem): a temperatura em si, e um sinal de controle que permite indicar a conversão desejada (de 1 a 6). Se o sinal de controle for 1, desejamos converter de Celsius para Fahrenheit; se for 2, desejamos converter de Celsius para Kelvin; se for 3, desejamos converter de Fahrenheit para Celsius; se for 4 de Fahrenheit para Kelvin; se for 5, desejamos converter de Kelvin para Celsius; se for 6 de Kelvin para Fahrenheit. A função deve retornar o valor convertido. Se o sinal de controle for diferente dos valores permitidos, a função não deve retornar nada (simplesmente não há o else). Para resolver essa questão, considere as seguintes relações entre grandezas de temperatura:

$$\frac{(T_{Fahrenheit} - 32)}{9} = \frac{T_{Celsius}}{5}$$

$$T_{Kelvin} - 273 = T_{Celsius}$$

$$\frac{(T_{Kelvin} - 273)}{5} = \frac{T_{Fahrenheit} - 32}{9}$$

4. Em sistemas digitais, todo tipo de informação é representado por meio de vetores binários, os quais correspondem a sequência de dígitos que podem assumir dois valores possíveis: 0 ou 1. Um exemplo de vetor binário de 8 dígitos é o seguinte: 00110010. Dois dispositivos digitais se comunicam enviando vetores binários de um transmissor até um receptor por meio de um canal de transmissão. É comum ocorrerem distorções causadas pelo canal de transmissão, as quais alteram a informação original enviada pelo

transmissor, fazendo chegar uma mensagem diferente ao receptor. Uma forma simples de checar a integridade da mensagem original consiste em adicionar um dígito extra no vetor binário original, que indica a quantidade de dígitos de valor "1" ou "0" presentes no vetor binário original. Esse dígito extra recebe o nome de "bit de paridade". Dessa forma, checando o vetor binário e o bit de paridade recebidos no receptor, é possível saber se houve distorção na mensagem original ou não.

Escreva uma função em Python que realiza a checagem de distorções no vetor binário recebido por um equipamento digital, através do método de paridade descrito anteriormente. A sua função deve receber 9 argumentos inteiros, os quais assumem apenas valores 0 ou 1. Os oito primeiros argumentos devem representar o vetor binário recebido e o nono argumento deve representar o bit de paridade que foi enviado junto com o vetor binário original. Para essa questão, assuma que se o bit de paridade valer "1", então deveria haver um número par de dígitos "1" presentes no vetor original. Do contrário, se o bit de paridade valer "0", então deveria haver um número ímpar de dígitos "1" presentes no vetor original. A função deve retornar o valor booleano "True" caso não haja distorções no vetor original (ou seja, a informação do bit de paridade, descrita anteriormente, coincide com o que foi observado no vetor binário representado pelos oito primeiros argumentos), e deve retornar "False" caso contrário.

5. Escreva uma função em Python que recebe por argumento três números representando posições em uma reta, e que retorne a maior distância encontrada entre dois pontos do conjunto recebido. Lembre-se que a distância sempre é um número positivo, mas as posições podem assumir valores positivos ou negativos. Por exemplo: a distância entre 3 e 5 é 2, assim como a distância entre 1 e 3. Outro exemplo: a distância entre -9 e -6 é 3, assim como a distância entre -12 e -9.
6. (Inspirada na OBI 2010 – Nível 1, Fase 1) A empresa local de abastecimento de água, a Saneamento Básico da Cidade (SBC), está promovendo uma campanha de conservação de água, distribuindo cartilhas e promovendo ações demonstrando a importância da água para a vida e para o meio ambiente. Para incentivar mais ainda a economia de água, a SBC alterou os preços de seu fornecimento de forma que, proporcionalmente, aqueles clientes que consumirem menos água paguem menos pelo metro cúbico. Todo cliente paga mensalmente uma assinatura de R\$ 10, que inclui uma franquia de 15 m³ de água. Isto é, para qualquer consumo entre 0 e 15 m³, o consumidor paga a mesma quantia de R\$ 10 reais (note que o valor da assinatura deve ser pago mesmo que o consumidor não tenha consumido água). Acima de 15 m³, cada metro cúbico subsequente tem um valor diferente, dependendo da faixa de consumo. A SBC cobra apenas por quantidades inteiras de metros cúbicos consumidos. A tabela abaixo especifica o preço por metro cúbico para cada faixa de consumo:

Faixa de consumo (m ³)	Preço (por m ³)
Até 15	Incluído na franquia
16 a 40	R\$ 1.2
41 em diante	R\$ 3.0

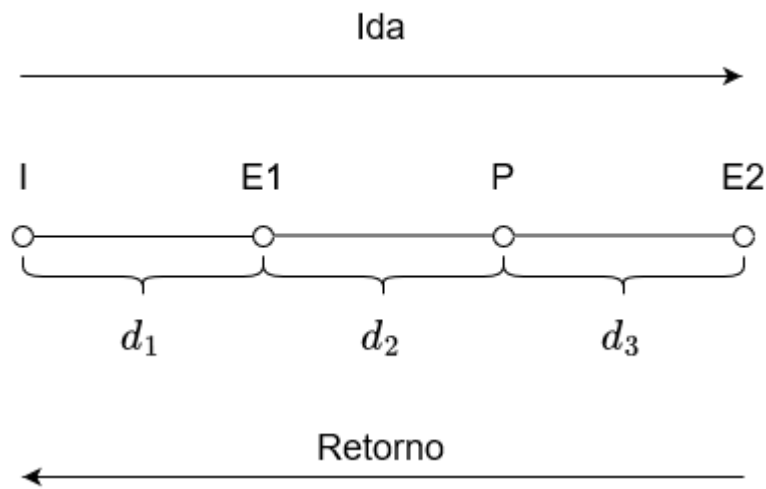
Por exemplo, se o consumo foi de 55 m³, o valor da conta é dado por:

- 10 reais da assinatura básica;
- 30 reais pelo consumo no intervalo 16 – 40 m³;
- 45 reais pelo consumo no intervalo 41 a 55 m³;

Isto é: $10 + 25 \cdot 1.2 + 15 \cdot 3.0 = \text{R\$ } 85.0$.

Escreva uma função em Python que recebe por argumento um inteiro igual ao valor do consumo de água uma residência, em m³, e retorna o valor da conta de água, a quantidade paga na segunda faixa, e a quantidade paga na terceira faixa. No caso do exemplo acima, o valor de retorno deveria ser (85.0, 30.0, 45.0).

7. Considere a situação em que um veículo precisa realizar 2 entregas em pontos diferentes da cidade, partindo de uma central de distribuição de produtos. Entretanto, o veículo dispõe de combustível limitado para realizar as entregas. O trajeto que ele segue, necessariamente, é exibido na figura abaixo.



O ponto "I" indica a central de distribuição dos produtos, de onde o veículo sempre parte com um valor inicial de combustível L_0 no tanque (assuma que o tanque cheio apresenta 60 litros). Os pontos "E" na figura simbolizam os locais de entrega. Entre os pontos de entrega, o veículo sempre passa por um posto de abastecimento, identificado pela letra "P" na figura, que o permite encher o tanque de combustível. Toda vez que ele passa pelo posto, ele enche o tanque com um valor fixo de L litros de combustível, não podendo exceder a capacidade máxima total do tanque de 60 litros. A entrega é considerada possível caso o veículo tenha combustível suficiente para passar pelos dois pontos de entrega e retornar para a central de distribuição, respeitando o trajeto indicado (primeiro passando pelo ponto de entrega 1, depois indo para o posto, e depois indo para o ponto de entrega 2). O trajeto de retorno é feito passando exatamente pelos mesmos pontos do trajeto de ida (parando, inclusive, no posto para reabastecer), conforme indicado na figura.

Escreva uma função em Python que receba, nesta ordem, os seguintes argumentos: a distância entre a central e o ponto de entrega 1 (em quilômetros), a distância entre o ponto de entrega 1 e o posto de abastecimento (em quilômetros), a distância entre o posto de abastecimento e o ponto de entrega 2 (em quilômetros), o consumo de combustível do veículo (em litros por quilômetro), a quantidade L_0 (em litros) que o veículo tem inicialmente no tanque, e a quantidade L (em litros) que o veículo reabastece de combustível ao passar pelo posto. A função deve retornar três valores, nesta ordem: um booleano, indicando se o veículo consegue realizar as entregas e retornar à central de distribuição (True) ou não (False), a quantidade de entregas que foram feitas, e a quantidade restante de combustível no tanque após o veículo retornar à central de distribuição (note que este valor será 0 nos casos em que o veículo não conseguir retornar à central de distribuição).

Antes de começar o código, é necessário verificar se os argumentos passados para a função são válidos. Todas as distâncias devem ser inteiros ou floats positivos não-nulos. Se alguma distância não atender a esse critério, a função deve retornar -1. A quantidade de combustível consumida pelo veículo por quilômetro, a quantidade L_0 inicial e a quantidade L de reabastecimento devem, também, ser inteiros ou floats positivos não-nulos. Caso isso não seja atendido em ao menos um desses argumentos, a função deve retornar -2. A quantidade L_0 inicial no tanque não deve ser superior à capacidade total do tanque (60 litros). Caso isso não seja respeitado, a função deve retornar -3. Verifique o arquivo de testes para exemplos de entrada e saída desta função.