Lista de Exercícios

1. Sabe-se que uma transformação linear é dada por $M = T^{-1} \cdot R \cdot T$. As matrizes **T** e **R** são:

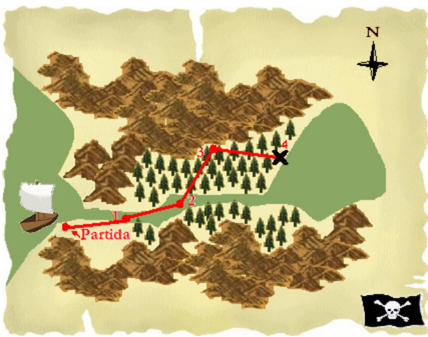
$$T = \begin{bmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \cos(\theta) & -\operatorname{sen}(\theta) & 0 \\ \operatorname{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Elabore um programa que solicita ao usuário os valores **Δx**, **Δy** e **θ** (em graus), calcula e exibe a matriz de transformação **M**.
Dica: crie as matrizes **T** e **R** como matrizes identidade de ordem 3. Depois altere os valores necessários.

2. Elabore um programa que verifica se três vetores são linearmente dependentes ou linearmente independentes. Crie a função **`LerVetor`** que preenche vetor **v** passado por parâmetro de referência com os valores digitados pelo usuário. Não se esqueça de criar os três vetores com zeros no programa principal antes de fazer a chamada da função de leitura.

3. Elabore um programa para calcular a soma de duas matrizes. O programa deve solicitar que o usuário digite o número de linhas e de colunas das matrizes, além dos valores pertencentes às matrizes. Crie uma função chamada **`LerMatriz`** que recebe como parâmetro de referência uma matriz e a retorna com os valores preenchidos com os números digitados pelo usuário. Não se esqueça de criar as matrizes (com as dimensões corretas) com zeros no programa principal, antes de fazer a chamada da função de leitura.

4. Você encontrou o mapa que indica o local do tesouro do pirata Shan Tee Lee e deve criar um programa para calcular a distância que será percorrida da partida (considere como a origem do sistema de coordenadas) até o local do tesouro, bem como as coordenas indicada pelo **X** no mapa.
O programa deve solicitar ao usuário o número de trechos, bem como as distâncias (em metros) e as direções (N, NE, E, SE, S, SO, O ou NO) de cada trecho.



Dica: crie um dicionário cujas chaves são as direções e os valores, os ângulos: 0, 45, 90, 135, 180, 225, 270 e 315, respectivamente. Cuidado! Note que a direção norte é o ângulo 0, assim, ao fazer a decomposição de cada vetor de deslocamento, utilize:

$$x = \text{distância} \cdot \operatorname{sen}(\text{ângulo}) \qquad y = \text{distância} \cdot \cos(\text{ângulo})$$

5. (Adaptado de Atividade de Geometria Analítica) Cerro Azul possui duas universidades: Alfa e Beta. As matrizes **A** e **B** representam o número de alunos matriculados nos cursos de graduação (G) e pós-graduação (P) de Alfa e Beta, separados por sexo (F) e (M).

$$A = \begin{array}{c} \begin{array}{cc} (G) & (P) \end{array} \\ \begin{bmatrix} 1500 & 300 \\ 2000 & 350 \end{bmatrix} \end{array} \begin{array}{c} \\ (F) \\ (M) \end{array} \qquad B = \begin{array}{c} \begin{array}{cc} (G) & (P) \end{array} \\ \begin{bmatrix} 1000 & 150 \\ 1500 & 190 \end{bmatrix} \end{array} \begin{array}{c} \\ (F) \\ (M) \end{array}$$

Elabore um programa que permita ao usuário informar os custos semestrais para a graduação e pós-graduação e responder às seguintes questões a partir de cálculo matricial.

a) Quantas mulheres estão matriculadas no curso de graduação em Cerro Azul?

b) Em um ano atípico, o número de alunos de Alfa dobrou e o de Beta foi reduzido à metade do valor original. Neste cenário, quais as novas matrizes A_1 e B_1? Repita o item anterior.

c) Qual a receita da universidade Alfa, gerada pelas mulheres, no cenário do item anterior? E pelos homens?

6. Uma empresa está projetando uma montanha russa e precisa de um programa para calcular o comprimento de cada trecho para determinar a quantidade de trilhos que deverá ser instalada. Sabe-se que o trecho estudado é uma descida e suas coordenadas são dadas pela seguinte equação paramétrica, no intervalo $0 \le t \le 4$:

$$\begin{cases} x(t) = t \\ y(t) = 11 + 9 \cdot \cos\left(\dfrac{\pi \cdot t}{4}\right) \end{cases}$$

O comprimento de uma curva é dado por: $L = \int_{t_i}^{t_f} \sqrt{(x')^2 + (y')^2}$.

Essa equação é válida para um domínio contínuo, mas para resolver o exercício, crie uma variável **`t`**, no intervalo informado, contendo muitos valores intermediários. As derivadas podem ser aproximadas pela função **`diff`**, enquanto que a integral pode ser aproximada como a soma dos valores de um array, dado pelo método **`sum`**.

7. A posição de um ponto material que se desloca ao longo de uma reta por 10 s é definida por $x(t) = -t^3 + 6 \cdot t^2 + 15 \cdot t + 40$, sendo **`x`** expresso em metros e **`t`** em segundos. Elabore um programa que calcule e exiba:

a) o instante no qual a velocidade será nula.

Dica: calcule a velocidade **v** como sendo **`dx/dt`** (utilizando **`diff`** para os valores de **x** e de **t**). Em seguida percorra o array **v** e verifique quando o valor subsequente ao observado cruza o eixo, ou seja, **`v[i]*v[i+1] < 0`**. Se isso acontecer, armazene o valor do índice **`i`** na variável **`iVo`**. O instante de tempo em que isso ocorre é a média entre **`t[iVo]`** e **`t[iVo+1]`**;

b) a posição da partícula nesse instante. Considere a média entre **`x[iVo]`** e **`x[iVo+1]`**;

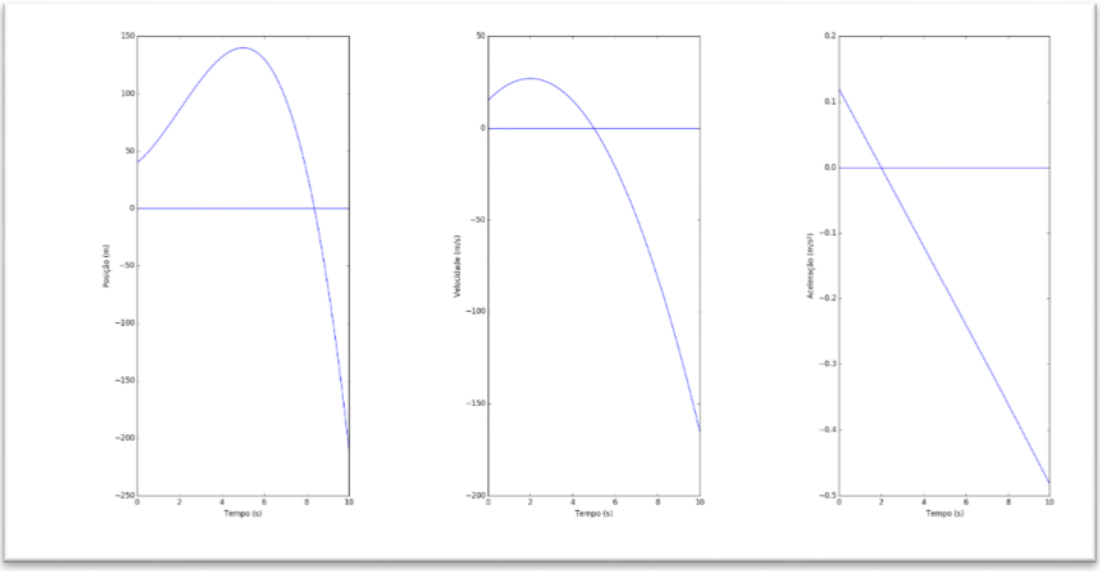
c) a aceleração da partícula nesse instante. Considere a média entre **`a[iVo]`** e **`a[iVo+1]`**;

Dica: calcule a aceleração a como sendo **`dv/dt[:-1]`** (utilizando **`diff`** para os valores de **v**).

d) os três gráficos, lado a lado.

Dica: para o gráfico da posição, utilize **`plot(t,x)`**;
para o gráfico da velocidade, utilize **`plot(t[-1:],v)`**;
para o gráfico da aceleração, utilize **`plot(t[-2:],a)`**.

Isso acontece por causa do **`diff`**, que gera um array com um elemento a menos do que o array original. Veja o resultado esperado dos gráficos.



Desafio:

Deseja-se implementar um sistema para representar um objeto tridimensional em uma imagem 2D. A variável **Modelo** contém as informações sobre as coordenadas dos vértices (primeira linha representa os valores de x, a segunda y e a terceira z) e a variável **Faces** representa os índices (colunas) dos vértices do **Modelo** (note que existem 6 faces definidas por 4 vértices cada).

$$\text{Modelo} = \begin{bmatrix} -1 & +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ -1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \end{bmatrix} \qquad \text{Faces} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 0 & 1 & 5 & 4 \\ 1 & 2 & 6 & 5 \\ 2 & 3 & 7 & 6 \\ 3 & 0 & 4 & 7 \end{bmatrix}$$

Inicialmente, considere que o observador está na posição $\mathrm{VRP} = \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$ e a distância focal vale $d = 0{,}8$. Sabe-se que o vetor $\mathrm{VUP} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

Para o cálculo da projeção, deve-se multiplicar algumas matrizes: $N_{per} = P_{per} \cdot R \cdot T$. Lembre-se que a multiplicação de matrizes no Python utiliza o operador `@`.

A matriz de projeção perspectiva é dada por:

$$P_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 2 \end{bmatrix}$$

A matriz de rotação é composta pelo cálculo de alguns vetores:

$$Rz = \frac{\mathrm{VRP}}{\|\mathrm{VRP}\|} \qquad Rx = \frac{\mathrm{VUP} \times Rx}{\|\mathrm{VUP} \times Rx\|} \qquad Ry = Rz \times Rx \qquad R = \begin{bmatrix} Rx_0 & Rx_1 & Rx_2 & 0 \\ Ry_0 & Ry_1 & Ry_2 & 0 \\ Rz_0 & Rz_1 & Rz_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de translação considera a posição o observador:

$$T = \begin{bmatrix} 1 & 0 & 0 & -VRP_0 \\ 0 & 1 & 0 & -VRP_1 \\ 0 & 0 & 1 & -VRP_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Após multiplicar as matrizes para obter N_{per}, divida todos os elementos da matriz pelo último elemento $\left(N_{per}[3,3]\right)$, garantindo que o último elemento torne-se 1.

Em seguida, calcule as coordenadas da projeção do **Modelo**, multiplicando $N_{per} \cdot \text{Modelo}$. Para cada coluna dos pontos projetados, divida todos as coordenadas das linhas pelo último elemento da respectiva coluna.

Sabe-se que a primeira linha da matriz projetada representa os valores de **`x`** da imagem 2D, enquanto que a segunda linha contém as informações de **`y`**. Separe os dados nessas duas variáveis.

Para exibir a imagem 2D, desenhe as faces utilizando a função **`fill`**, com transparência de 10%. Utilize o seguinte algoritmo:

> Para cada face **`f`** das **`Faces`**:
>
>   Crie uma lista vazia para a componente X outra para o Y dos vértices;
>
>   Para cada vértice **`v`** da face **`f`**:
>
>   Adicione à lista de vértices de X o valor de **x[v]**;
>
>   Adicione à lista de vértices de Y o valor de **y[v]**;
>
>   Desenhe a face passando a lista de vértices de X e de Y;

Não se esqueça de manter a proporção dos eixos iguais.