

# Programación

## Guía 5

10 de Octubre 2019

**Problema 1:** Dados los pesos de  $n = 20$  cajas a transportar en un camión guardado en un archivo `pesos.txt` y el peso máximo que puede transportar el camión  $pmax = 200$ .

- (a) Leer los pesos de `pesos.txt` y asignar a un array.
- (b) Determinar si el camión puede transportar las cajas usando `.sum`.
- (c) Determinar en una función cuantas cajas tienen el peso mayor al peso promedio del grupo de cajas usando `.mean`.
- (d) En otra función vea en el caso en que el peso de las cajas exceda el peso máximo que puede transportar el camión elimine las cajas que sean necesarias comenzando por las mas pesadas `.max` o `.argmax`. Guarde en un archivo `seleccionadas.txt` las cajas se pueden subir al camión. Utilice en todo momento arrays (no listas)

**Problema 2:** Implemente un programa que reciba un vector con posiciones en el espacio y las refleje respecto del plano  $y - z$

**Problema 3:** Se puede rotar un punto alrededor del eje  $z$  mediante la siguiente transformación

$$\mathbf{r}' = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (1)$$

donde  $\mathbf{r}'$  es el vector (matrix columna) cuyas componentes rotadas son  $(x', y', z')$ . Implementar un programa que dado un vector  $\mathbf{r}$  con sus componentes en el espacio, de obtenga el vector rotado alrededor del eje  $z$  en un ángulo  $\theta$ .

Por otro lado, la traslación de la posición en la dirección  $z$  se puede implementar como

$$\mathbf{r}' = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ D \end{pmatrix} \quad (2)$$

Asumiendo que  $\theta = \omega t$  (movimiento circular uniforme) y que  $D = Vt$  (MRU) implementar un programa que obtenga la trayectoria de la partícula en el espacio.

**Problema 4:** Considere el problema de fichas de las damas en un tablero. Respecto del borde del tablero las fichas tienen una ubicación  $i, j$  donde  $i$  indica la distancia horizontal a la que esta la ficha respecto del borde superior izquierdo del tablero y  $j$  indica la distancia en la dirección vertical (de arriba hacia abajo). Cada ficha posee un número entero que identifica a la misma.

1. Implemente una función que busque una ficha que posee un número particular dentro del conjunto, e indique como respuesta en que lugar del tablero se encuentra dicha ficha. Usando bucles (Sin funciones numpy).
2. Realice el item anterior con funciones numpy.
3. Realice una función distancia (en unidades de ubicaciones) que ingresen dos números de fichas y salga la distancia a la que se encuentran las fichas.

**Problema 5:** Implementar un programa que dada una matriz de  $n$  por  $n$ , responda si la matriz es simétrica o no.

**Problema 6:** Implementar un programa que verifique si una matriz de dimensión  $n$  es ortogonal, es decir si el producto de ella por su traspuesta es la matriz identidad.

**Problema 7:** Genere 4 vectores de  $N = 10$  dimensiones.

1. Guarde estos vectores en un archivo vectores.npy. Lea los vectores.
2. Con los tres primeros vectores construya una matriz tridiagonal (desechando la última componente de dos de estos vectores).
3. Implementar una función que realice por componentes el producto de una matriz tridiagonal  $N \times N$  por un vector de  $N$  dimensiones (el cuarto vector del archivo).

**Problema 8:** Implementar un programa que dada una malla de enteros  $M_{i,j}$  de  $n \times n$ , obtenga la suma de los segundos vecinos de cualquier elemento  $i,j$ . Los segundos vecinos son los que se encuentran en las diagonales del cuadrado que se genera alrededor del punto de interés.

**Problema 9:** Realice un programa interactivo que permita jugar al tateti.

**F@CENA © 2019**