Tarea Nro. 3 - LinAlg + Sympy

- Nombre y apellido: Renato Balcázar
- Fecha: 01-12-2020

Producto punto

1. Use un producto punto y la lista de compras de la tabla 9.4 para determinar su cuenta total en la tienda.

Tabla 9.4 Lista de compras

Artículo	Número necesario	Costo
Leche	2 galones	$3.50 por galón
Huevos	1 docena	$1.25 por docena
Cereal	2 cajas	$4.25 por caja
Sopa	5 latas	$1.55 por lata
Galletas	1 paquete	$3.15 por paquete

In [3]:
```
import numpy as np

products = np.array([2,1,2,5,1])
prices = np.array([3.5, 1.25, 4.25, 1.55, 3.15])

result = products.dot(prices)
result
```

Out[3]:
```
27.65
```

Multiplicación matricial

1. Con un calorímetro de bomba se realizó una serie de experimentos. En cada experimento se usó una cantidad diferente de agua. Calcule la capacidad calorífica total para el calorímetro en cada uno de los experimentos, mediante multiplicación matricial, los datos de la tabla 9.8 y la información acerca de la capacidad calorífica que sigue a la tabla.

Tabla 9.8 Propiedades térmicas de un calorímetro de bomba

Experimento núm.	Masa de agua	Masa de acero	Masa de aluminio
1	110 g	250 g	10 g
2	100 g	250 g	10 g
3	101 g	250 g	10 g
4	98.6 g	250 g	10 g
5	99.4 g	250 g	10 g

Componente	Capacidad calorífica
Acero	0.45 J/gK
Agua	4.2 J/gK
Aluminio	0.90 J/gK

In [24]:
```
propierties = np.array([
    [ 110, 250, 10],
    [ 100, 250, 10],
    [ 101, 250, 10],
    [98.6, 250, 10],
    [99.4, 250, 10]
])

capacity_elements = np.array([
    [0.45],
    [4.20],
    [0.90]
])

res = np.dot(propierties, capacity_elements)
res
```

Out[24]:
```
array([[1108.5 ],
       [1104.  ],
       [1104.45],
       [1103.37],
       [1103.73]])
```

Determinantes e inversos

1. Recuerde que no todas las matrices tienen inverso. Una matriz es singular (es decir: no tiene inverso) si su determinante es igual a 0 (es decir, |A| = 0). Use la función determinante para probar si cada una de las siguientes matrices tiene inverso:

$$A = \begin{bmatrix} 2 & -1 \\ 4 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 2 \\ 5 & -4 & 0 \end{bmatrix}$$

Si existe un inverso, calcúlelo.

In [23]:
```
A = np.array([
    [2, -1],
    [4,  5]
])

B = np.array([
    [4, 2],
    [2, 1]
])

C = np.array([
    [ 2,  0,  0],
    [ 1,  2,  2],
    [ 5, -4,  0]
])

if (np.linalg.det(A) != 0):
    print("\nInversa de A: ")
    print(np.linalg.inv(A))
else:
    print("\nA no posee una inversa porque su determinante es 0")

if (np.linalg.det(B) != 0):
    print("\nInversa de B: ")
    print(np.linalg.inv(B))
else:
    print("\nB no posee una inversa porque su determinante es 0")

if (np.linalg.det(C) != 0):
    print("\nInversa de C: ")
    print(np.linalg.inv(C))
else:
    print("\nC no posee una inversa porque su determinante es 0")
```

```
Inversa de A:
[[ 0.35714286  0.07142857]
 [-0.28571429  0.14285714]]

B no posee una inversa porque su determinante es 0

Inversa de C:
[[ 0.5    0.     0.   ]
 [ 0.625  0.    -0.25 ]
 [-0.875  0.5    0.25 ]]
```

Resolución de sistemas ecuaciones lineales ¶

1. Resuelva el siguiente sistema de ecuaciones

$$\begin{aligned}
3x_1 + 4x_2 + 2x_3 - x_4 + x_5 + 7x_6 + x_7 &= 42 \\
2x_1 - 2x_2 + 3x_3 - 4x_4 + 5x_5 + 2x_6 + 8x_7 &= 32 \\
x_1 + 2x_2 + 3x_3 + x_4 + 2x_5 + 4x_6 + 6x_7 &= 12 \\
5x_1 + 10x_2 + 4x_3 + 3x_4 + 9x_5 - 2x_6 + x_7 &= -5 \\
3x_1 + 2x_2 - 2x_3 - 4x_4 - 5x_5 - 6x_6 + 7x_7 &= 10 \\
-2x_1 + 9x_2 + x_3 + 3x_4 - 3x_5 + 5x_6 + x_7 &= 18 \\
x_1 - 2x_2 - 8x_3 + 4x_4 + 2x_5 + 4x_6 + 5x_7 &= 17
\end{aligned}$$

In [4]:
```
coef = np.array([
    [ 3,  4,  2, -1,  1,  7,  1],
    [ 2, -2,  3, -4,  5,  2,  8],
    [ 1,  2,  3,  1,  2,  4,  6],
    [ 5, 10,  4,  3,  9, -2,  1],
    [ 3,  2, -2, -4, -5, -6,  7],
    [-2,  9,  1,  3, -3,  5,  1],
    [ 1, -2, -8,  4,  2,  4,  5]
])
results = np.array([
    42,
    32,
    12,
    -5,
    10,
    18,
    17
])


respuesta = np.linalg.solve(coef, results)
respuesta
```

Out[4]:
```
array([-0.18899493,  2.54589061, -3.28057396, -6.75778176,  1.32124449,
        4.31944831,  0.62940585])
```

Cálculo

1. La capacidad calorífica C_p de un gas se puede modelar con la ecuación empírica

$$C_p = a + bT + cT^2 + dT^3$$

donde a, b, c y d son constantes empíricas y T es la temperatura en grados Kelvin. El cambio en entalpía (una medida de energía) conforme el gas se caliente de T_1 a T_2 es la integral de esta ecuación con respecto a T:

$$\triangle\, h = \int_{T_1}^{T_2} C_p\, dT$$

Encuentre el cambio en entalpía del oxígeno gaseoso conforme se calienta de 300 K a 1000 K. Los valores de a, b, c y d para el oxígeno son

$$\begin{aligned} a &= 25.48 \\ b &= 1.520x10^{-2} \\ c &= -0.7155x10^{-5} \\ d &= 1.312x10^{-9} \end{aligned}$$

In [17]:
```
from sympy import *
t = Symbol('t')

t_1 = 300
t_2 = 1000

a = 25.48
b = 1.520*(10**(-2))
c = -0.7155*(10**(-5))
d = 1.312*(10**(-9))


capacity = a + b*t + c*(t**2) + d*(t**3)

h_variation = integrate(capacity, (t, t_1, t_2))
h_variation
```

Out[17]:
```
22756.7382
```