In [ ]:
```
import pandas as pd
import numpy as np
tabela_dieta_csv = pd.read_csv('dietaDados.csv')
tabela = tabela_dieta_csv.T
tabela
```

Out[ ]:

	0	1	2	3	4	5	6	7	8	9	...	11	12	13	14	15	16	17	18	19	20
Mercadoria	Farinha de trigo (enriquecida)	Farinha de milho	Leite evaporado (lata)	Margarina	Queijo (cheddar)	Pasta de amendoim	Bacon	Fígado (boi)	Lombo de porco assado	Salmão Rosa	...	Repolho	Cebola	Batatas	Espinafre	Batata0doce	Pêssegos secos	Ameixas secas	Feijão verde seco	Feijão branco seco	Produto X
Calorias(1000)	44.0	36.0	8.4	20.6	7.4	15.7	41.7	2.2	4.4	5.8	...	2.6	5.8	14.3	1.1	9.6	8.5	12.8	17.4	26.9	58.0
Proteinas(gramas)	7	897	422	17	448	661	0	333	249	705	...	125	166	336	106	138	87	99	1055	1691	0
Cálcio(gramas)	1411.0	1.7	15.1	0.6	16.4	1.0	0.0	0.2	0.3	6.8	...	4.0	3.8	1.8	0.0	2.7	1.7	2.5	3.7	11.4	6.3
Ferro(mg)	2.0	99.0	9.0	6.0	19.0	48.0	0.0	139.0	37.0	45.0	...	36.0	59.0	118.0	138.0	54.0	173.0	154.0	459.0	792.0	0.0
Vitamina A(1000I.U.)	365.0	30.9	26.0	55.8	28.1	0.0	0.2	169.2	0.0	3.5	...	7.2	16.6	6.7	918.4	290.7	86.8	85.7	5.1	0.0	52.0
Tiamina(mg.)	0.0	17.4	3.0	0.2	0.8	9.6	0.0	6.4	18.2	1.0	...	9.0	4.7	29.4	5.7	8.4	1.2	3.9	26.9	38.4	0.0
Riboflavina(mg.)	55.4	7.9	23.5	0.0	10.3	8.1	5.0	50.8	3.6	4.9	...	4.5	5.9	7.1	13.8	5.4	4.3	4.3	38.2	24.6	4.4
Niacina(mg.)	33.3	106.0	11.0	0.0	4.0	471.0	5.0	316.0	79.0	209.0	...	26.0	21.0	198.0	33.0	83.0	65.0	65.0	93.0	217.0	0.0
Ácido Ascórbico(mg.)	441	0	60	0	0	0	0	525	0	0	...	5369	1184	2522	2755	1912	257	257	0	0	45

10 rows × 21 columns

Segundo as iniciais do meu nome, a seguinte distribuição de nutrientes se apresenta:

Para a letra F, teremos 5.9 calorias de restrição, para a letra A, teremos 0.07 gramas de cálcio, 6 unidades internacionais de vitamina A, para a letra I, teremos 6.5 riboflavina e 10 gramas de ácido ascórbico.

Para as demais vitaminas e nutrientes, teremos uma restrição de maior igual a zero.

Para utilizar a rotina linprog da biblioteca SciPy, precisamos nos comportar segundo especificação. Não há inequação de maior ou igual, apenas de menor ou igual e igualdade (a_ub, b_ub, a_eq, b_eq). A primeira dupla representa a matriz e vetor de inequações de restrições e a segundo matriz e vetor de igualdades de restrições.

Com isso, todas as entradas da matriz de restrições e o vetor de restrições precisam ter seu valor invertido devido ao fato de queremos maior ou igual a zero, não o contrário, que é a forma que a rotina espera.

In [ ]:
```
restricoes = np.array([-5.9,0.0,-.07,0.0,-6.0,0.0,-6.5,0.0,-10.0])
restricoes
```

Out[ ]:
```
array([ -5.9 ,   0.  ,  -0.07,   0.  ,  -6.  ,   0.  ,  -6.5 ,   0.  ,
       -10.  ])
```

Tranformando a tabela em matriz de restrições e transformando os valores em negativo para continuar a implementação da especificação.

In [ ]:
```
matriz = tabela.drop('Mercadoria').to_numpy()

for i in range(len(matriz)):
    matriz[i] = -matriz[i]

matriz
```

Out[ ]:
```
array([[-44.0, -36.0, -8.4, -20.6, -7.4, -15.7, -41.7, -2.2, -4.4, -5.8,
        -2.4, -2.6, -5.8, -14.3, -1.1, -9.6, -8.5, -12.8, -17.4, -26.9,
        -58.0],
       [-7, -897, -422, -17, -448, -661, 0, -333, -249, -705, -138, -125,
        -166, -336, -106, -138, -87, -99, -1055, -1691, 0],
       [-1411.0, -1.7, -15.1, -0.6, -16.4, -1.0, -0.0, -0.2, -0.3, -6.8,
        -3.7, -4.0, -3.8, -1.8, -0.0, -2.7, -1.7, -2.5, -3.7, -11.4,
        -6.3],
       [-2.0, -99.0, -9.0, -6.0, -19.0, -48.0, -0.0, -139.0, -37.0,
        -45.0, -80.0, -36.0, -59.0, -118.0, -138.0, -54.0, -173.0,
        -154.0, -459.0, -792.0, -0.0],
       [-365.0, -30.9, -26.0, -55.8, -28.1, -0.0, -0.2, -169.2, -0.0,
        -3.5, -69.0, -7.2, -16.6, -6.7, -918.4, -290.7, -86.8, -85.7,
        -5.1, -0.0, -52.0],
       [-0.0, -17.4, -3.0, -0.2, -0.8, -9.6, -0.0, -6.4, -18.2, -1.0,
        -4.3, -9.0, -4.7, -29.4, -5.7, -8.4, -1.2, -3.9, -26.9, -38.4,
        -0.0],
       [-55.4, -7.9, -23.5, -0.0, -10.3, -8.1, -5.0, -50.8, -3.6, -4.9,
        -5.8, -4.5, -5.9, -7.1, -13.8, -5.4, -4.3, -4.3, -38.2, -24.6,
        -4.4],
       [-33.3, -106.0, -11.0, -0.0, -4.0, -471.0, -5.0, -316.0, -79.0,
        -209.0, -37.0, -26.0, -21.0, -198.0, -33.0, -83.0, -65.0, -65.0,
        -93.0, -217.0, -0.0],
       [-441, 0, -60, 0, 0, 0, 0, -525, 0, 0, -862, -5369, -1184, -2522,
        -2755, -1912, -257, -257, 0, 0, -45]], dtype=object)
```

Chamando a rotina Linprog da biblioteca SciPy e escolhendo o método revised simplex para resolver o sistema linear

In [ ]:
```
import numpy as np
from scipy.optimize import linprog

c = np.array([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1])
A = matriz
B = restricoes

res = linprog(c, A_ub=A, b_ub=B,bounds=(0, None))
print(res)
```

```
     con: array([], dtype=float64)
     fun: 0.12978540205420117
 message: 'Optimization terminated successfully.'
     nit: 13
   slack: array([6.13020745e-11, 8.13776670e-01, 1.63419373e+02, 2.32507620e-01,
       3.71362830e+01, 2.48239304e-10, 1.51996637e-10, 3.87125182e+00,
       4.18768512e+01])
  status: 0
 success: True
       x: array([1.16253808e-01, 2.65077906e-12, 6.96911807e-13, 9.47512776e-13,
       7.09038928e-13, 9.17090676e-13, 2.28722374e-12, 7.71428061e-13,
       7.16349952e-13, 6.89554781e-13, 6.59909511e-13, 6.69459923e-13,
       7.04415705e-13, 9.72871729e-13, 6.68118668e-13, 7.71343614e-13,
       7.18193807e-13, 7.94367379e-13, 1.68710310e-12, 1.85920603e-12,
       1.35315936e-02])
```

Descobrindo a dieta.

O custo total foi de 0.12978540205420117 centavos.

Como foi descoberto, usaremos todos os itens da lista do mercado, distribuído da seguinte forma:

1.16253808e-01 Farinha de trigo (enriquecida),

2.65077906e-12 Farinha de milho,

6.96911807e-13 Leite evaporado (lata),

9.47512776e-13 Margarina,

7.09038928e-13 Queijo(Cheddar),

9.17090676e-13 Pasta de amendoim,

2.28722374e-12 Bacon ,

7.71428061e-13 Figado(boi),

7.16349952e-13 Lombo de porco assado,

6.89554781e-13 Salmão rosa,

6.59909511e-13 Feijão verde,

6.69459923e-13 Repolho,

7.04415705e-13 Cebola,

9.72871729e-13 Batatas,

6.68118668e-13 Espinafre,

7.71343614e-13 Batata-doce,

7.18193807e-13 Pêssegos,

7.94367379e-13 Ameixas secas,

1.68710310e-12 Feijão verde,

1.85920603e-12 Feijão branco,

1.35315936e-02 Produto X