Wong Zhing Jieh Jack   201910656

1. The table below summarizes the model of my food intake.

   Meaning of variables:
   X1: 1 slice of salmon, 110g
   X2: 0.5 cup of broccoli, 45g
   X3: 1 egg, 60g
   X4: 0.5 cup sliced carrot, 25g
   X5: 1 slice wholemeal bread, 50g
   X6: 1 cup brown rice, 100g (nutritional value taken at uncooked 100g)

	X1	X2	X3	X4	X5	X6
Pris (kr)/serving	27.5	4.5	2	0.4	1	1.5
Eng frm Protein(kj)	23.2g*17kj/g = 394.4	1.08g*17kj/g = 18.36	49g*17kj/g=833	0.28g*17kj/g = 4.76	4g*17kj/g=68	9g*17kj/g= 153
Eng frm Carbs(kj)	0	1.8g*17kj/g= 30.6	4g*17kj/g= 68	1.18g*17kj/g = 20.06	23g*17kj/g= 391	70g*17kj/g = 1190
Eng frm Fats(kj)	2.53g*38kj/g = 96.14	0.18g*38kj/g = 6.84	0	0.05g*38kj/g = 1.9	1.3g*38kj/g = 49.4	2.9g*38kj/g = 110.2
Total Eng per serving	490.54	25.8	901	26.72	508.4	1453.2

Figure 1: Macronutrients and prices for respective food

The objective function, or the price, is **27.5X1 + 4.5X2 + 2X3 + 0.4X4 + X5 + X6**, which is to be minimized, subject to the constraints:

Since we need at least 10000kj to be the total energy intake, we have
490.54 X1 + 25.8 X2 + 901 X3 + 26.72 X4 + 508.4 X5 + 1453.2 X6 >= 10000

subject to

Proportion of energy from proteins:
0.25 <= (394.4X1 + 18.36X2 + 833X3 + 4.76X4 + 68X5 + 153X6)/(490.54 X1 + 25.8 X2 + 901 X3 + 26.72 X4 + 508.4 X5 + 1453.2 X6) <= 0.4

Resulting in two eqns:
271.765 X1 + 11.91 X2 + 607.75 X3 - 1.92 X4 - 59.1 X5 - 210.3 X6 >= 0

198.184 X1 + 8.04 X2 + 472.6 X3 + -5.928 X4 -135.36 X5 - 428.28 X6 <= 0

Proportion of energy from carbs:
0.45 <= (0.6X2 + 68X3 + 20.06X4 + 391X5 + 1190X6)/(490.54 X1 + 25.8 X2 + 901 X3 + 26.72 X4 + 508.4 X5 + 1453.2 X6) <= 0.6

Resulting in two eqns:
-220.743 X1 – 11.01 X2 - 337.45 X3 + 8.036 X4 + 162.22 X5 + 536.06 X6 >= 0

-294.324 X1 – 14.88 X2 – 472.6 X3 + 4.028 X4 + 85.96 X5 + 318.08 X6 <= 0

Proportion energy from fats:
0.1 <= (96.14X1 + 6.84X2 + 1.9X3 + 49.4X5 + 110.2X6)/(490.54 X1 + 25.8 X2 + 901 X3 + 26.72 X4 + 508.4 X5 + 1453.2 X6) <= 0.2

Resulting in two eqns:
47.086 X1 + 4.26 X2 – 88.2 X3 –2.672 X4 – 1.44 X5 – 35.12 X6 >= 0

-1.968 X1 + 1.68 X2 – 178.3 X3 – 5.344 X4 + -52.28 X5 – 180.44 X6 <= 0

Additional constraints (personal preferences):
1 to 10 slices of salmon: 1 <= X1 <= 10
1 to 4 servings of broccoli: 1 <= X2 <= 4
1 to 4 eggs: 1 <= X3 <= 4
1 to 4 servings of sliced carrots: 1 <= X4 <= 4
1 to 10 slices of whole meal bread: 1 <= X5 <= 10
1 to 5 cups of brown rice: 1 <= X6 <= 5

2. The first step to modelling my diet is identifying the objective function, which is the price of the total daily food intake.
   The next step is to identify the decision variables of this model. One way is to make the weigh (grams) of every food item a decision variable. A friendlier approach in my opinion is to use serving of every food item a decision variable. For example, ½ cup of broccoli or 1 slice of bread.
   Next, I would break down a food item into its macro nutrient components (proteins, carbs, fats) and find out its corresponding energy contents.
   The next step which is crucial is identifying the constraints, such as the proportion of each macronutrient in contributing to the total daily energy consumption.
   Some additional constraints are applied to make it more reasonable and personal. For example, I limit myself to 10 slices of bread per day and at least one of each food item mentioned.

3. Using *scipy.optimize.linprog*, an optimal solution is found to be 110.31, which means the lowest price of my daily food is 110.31kr. The food consists of 3.3 slices of salmon, 1 serving (0.5 cup) of broccoli, 1 egg, 1 serving (0.5 cup) of sliced carrots, 10 slices of wholemeal bread, 1.6 cup of brown rice. This is a reasonable diet for me considering I will split this into breakfast, lunch and dinner.

Wong Zhing Jieh Jack 201910656

4. Below is the python code:

```
import scipy.optimize

c = [27.5, 4.5, 2, 0.4, 1, 1]
A = [
     [-490.54, -25.8, -901, -26.72, -508.4, -1453.2],
     [-271.765, -11.91, -607.75, 1.92, 59.1, 210.3],
     [198.184, 8.04, 472.6, -5.928, -135.36, -428.28],
     [220.743, 11.01, 337.45, -8.036, -162.22, -536.06],
     [-294.324, 14.88, -472.6, 4.028, 85.96, 318.08],
     [-47.086, -4.26, 88.2, 2.672, 1.44, 35.12],
     [-1.968, 1.68, -178.3, -5.344, -52.28, -180.44],
     [1, 0, 0, 0, 0, 0],
     [-1, 0, 0, 0, 0, 0],
     [0, 1, 0, 0, 0, 0],
     [0, -1, 0, 0, 0, 0],
     [0, 0, 1, 0, 0, 0],
     [0, 0, -1, 0, 0, 0],
     [0, 0, 0, 1, 0, 0],
     [0, 0, 0, -1, 0, 0],
     [0, 0, 0, 0, 1, 0],
     [0, 0, 0, 0, -1, 0],
     [0, 0, 0, 0, 0, 1],
     [0, 0, 0, 0, 0, -1]
]
b = [-10000, 0, 0, 0, 0, 0, 0, 10, -1, 4, -1, 4, -1, 4, -1, 10, -1, 5, -1]


print(scipy.optimize.linprog(c,A,b))
```

Output:

```
con: array([], dtype=float64)
     fun: 110.30851540840675
 message: 'Optimization terminated successfully.'
     nit: 9
   slack: array([2.90321623e-07, 5.97591707e+02, 9.02408293e+02, 1.40240829e+03,
       6.78317067e+01, 2.17586660e-09, 1.00000000e+03, 6.66150084e+00,
       2.33849916e+00, 3.00000000e+00, 5.56828139e-10, 3.00000000e+00,
       8.15192358e-12, 3.00000000e+00, 1.72533543e-10, 3.76559228e-10,
       9.00000000e+00, 3.40021152e+00, 5.99788482e-01])
  status: 0
 success: True
       x: array([ 3.33849916,  1.        ,  1.        ,  1.        , 10.        ,
        1.59978848])
```