Part 1: Chosen Foods

The following foods were chosen for the diet problem and are staple ingredients in my daily diet. The following values listed in the table are not per serving, but per 100 grams (which in some cases is equal to 1 serving).

Food	Sodium (mg)	Energy (kcal)	Protein (g)	Vitamin D (mcg)	Calcium (mg)	Iron (mg)	Potassium (mg)	Cost ($)
White Rice	1	130	2.69	0	10	0.2	35	0.12
Ground Beef (88/12)	58	187.5	19.6	0	12.5	2.7	0	1.05
Spinach	79	23	2.86	0	99	2.71	558	0.70
Large Egg	142	144	12.6	2	56	1.8	138	0.76
Banana	1	89	1	0	5	0.26	358	0.11

Part 2: Linear Programming Problem Variable Specifications

Plain English Explanation

In the following problem, we are trying to determine the most ideal diet that will meet daily/ weekly nutritional standards while keeping cost as low as possible. The possible foods that can be included in the diet are White Rice, Ground Beef, Spinach, Large Eggs, and Bananas.

Decision Variables

The following table represents all variables that can be modified to find the most optimal solution. Each variable identifier is arbitrary and is just meant to reduce the length of the other displayed equations.

Variable	Description
x_1	White Rice
x_2	Ground Beef
x_3	Spinach
x_4	Large Egg
x_5	Banana

Objective function with cost coefficients

Minimize: $Cost = 0.12x_1 + 1.05x_2 + 0.70x_3 + 0.76x_4 + 0.11x_5$

Daily nutritional constraints

Variable	Constraint Type	Value
Sodium (mg)	Maximum	5000
Energy (kcal)	Minimum	2000
Protein (g)	Minimum	50
Vitamin D (mcg)	Minimum	20
Calcium (mg)	Minimum	1300

Iron (mg)	Minimum	18
Potassium (mg)	Minimum	4700

Part 3: Solving the problem in Python using PuLP

Below is the solution I came up with using PuLP's *LpMinimize* function.

```
#Define the variables
WhiteRice = LpVariable("WhiteRice", 0, None)
GroundBeef = LpVariable("GroundBeef", 0, None)
Spinach = LpVariable("Spinach", 0, None)
LargeEgg = LpVariable("LargeEgg", 0, None)
Banana = LpVariable("Banana", 0, None)

#Define the problem (In this case, we want to minimize cost)
prob = LpProblem("Problem", LpMinimize)

#Define the Contraints
prob += 1 * WhiteRice    + 58 * GroundBeef    + 79 *  Spinach  + 142 * LargeEgg   + 1 * Banana <= 5000 #Sodium
prob += 130 * WhiteRice  + 187.5 * GroundBeef + 23 * Spinach   + 144 * LargeEgg   + 89 * Banana >= 2000 #Energy
prob += 2.69 * WhiteRice + 19.6 * GroundBeef  + 2.86 *Spinach  + 12.6 * LargeEgg + 1 * Banana >= 50 #Protein
prob += 0 * WhiteRice    + 0 * GroundBeef     + 0 * Spinach    +  2 * LargeEgg    + 0 * Banana >= 20 #Vitamin D
prob += 10 * WhiteRice   + 12.5 * GroundBeef  + 99 * Spinach   +  56 * LargeEgg   + 5 * Banana >= 1300 # Calcium
prob += 0.2 * WhiteRice  + 2.7 * GroundBeef   + 2.71 * Spinach +  1.8 * LargeEgg + 0.26 * Banana >= 18 #Iron
prob += 35 * WhiteRice   + 0 * GroundBeef     + 558 * Spinach +  138 * LargeEgg  + 358 * Banana >= 4700
#Potassium

#define the opjective function
prob += 0.12 * WhiteRice + 1.05 * GroundBeef    + 0.7 *  Spinach  + 0.76 * LargeEgg   + 0.11 * Banana


#Solve the problem
status = prob.solve()
print(f"Diet Problem")
print(f"Optimization Status={LpStatus[status]}\n")

# print the results
for variable in prob.variables():
    print(f"{variable.name} = {variable.varValue *100:.1f} grams")

print(f"\nDaily Cost = ${value(prob.objective):.2f}")
print(f"Weekly Cost = ${7 * value(prob.objective):.2f}")
print(f"")
```

This was the following output:

```
Diet Problem
Optimization Status=Optimal

Banana = 0.0 grams
GroundBeef = 0.0 grams
LargeEgg = 1000.0 grams
Spinach = 716.8 grams
WhiteRice = 304.0 grams

Daily Cost = $12.98
Weekly Cost = $90.88
```

The most notable things that stood out to me about these results was the fact that Ground Beef and Bananas were entirely excluded from the results. This is one of the main challenges with using Linear Programming to solve the Diet problem is that it will *always* take the most cost-efficient route. Ground Beef, while similar to the nutritional value of a Large Egg, is nearly and extra third of the cost meaning that it will always be more optimal to use Eggs compared to beef.

The main way that the two missing foods can be included in the solution is by further constraining the program and to provide a minimum quantity that should be included.

Part 4: Adding further constraints so all Foods are used

```
#Problem 2: Adding more constaints

#Define the variables
WhiteRice = LpVariable("WhiteRice", 0, None)
GroundBeef = LpVariable("GroundBeef", 0, None)
Spinach = LpVariable("Spinach", 0, None)
LargeEgg = LpVariable("LargeEgg", 0, None)
Banana = LpVariable("Banana", 0, None)

#Define the problem (In this case, we want to minimize cost)
prob = LpProblem("Problem", LpMinimize)

#Define the Contraints
prob += 1 * WhiteRice    + 58 * GroundBeef    + 79 *  Spinach  + 142 * LargeEgg   + 1 * Banana <=
5000 #Sodium
prob += 130 * WhiteRice  + 187.5 * GroundBeef + 23 * Spinach   + 144 * LargeEgg   + 89 * Banana >=
2000 #Energy
prob += 2.69 * WhiteRice + 19.6 * GroundBeef  + 2.86 *Spinach  + 12.6 * LargeEgg + 1 * Banana >=
50 #Protein
prob += 0 * WhiteRice    + 0 * GroundBeef     + 0 * Spinach    +  2 * LargeEgg    + 0 * Banana >=
20 #Vitamin D
prob += 10 * WhiteRice   + 12.5 * GroundBeef  + 99 * Spinach   +  56 * LargeEgg   + 5 * Banana >=
1300 # Calcium
prob += 0.2 * WhiteRice  + 2.7 * GroundBeef   + 2.71 * Spinach +  1.8 * LargeEgg + 0.26 * Banana
>= 18 #Iron
prob += 35 * WhiteRice   + 0 * GroundBeef     + 558 * Spinach +  138 * LargeEgg  + 358 * Banana >=
4700 #Potassium
prob += GroundBeef >= 2 # At least 200 grams of ground beef
prob += Banana >= 2 # At least 100 grams of banana (Approximately a single banana)
prob += WhiteRice >= 3 #At least 300 g of White

#define the opjective function
prob += 0.12 * WhiteRice + 1.05 * GroundBeef    + 0.7 *  Spinach  + 0.76 * LargeEgg   + 0.11 *
Banana


#Solve the problem
status = prob.solve()
print(f"Diet Problem")
print(f"Optimization Status={LpStatus[status]}\n")

# print the results
for variable in prob.variables():
    print(f"{variable.name} = {variable.varValue *100:.1f} grams")

print(f"\nDaily Cost = ${value(prob.objective):.2f}")
print(f"Weekly Cost = ${7 * value(prob.objective):.2f}")
print(f"")
```

This was the following result:

```
Diet Problem
Optimization Status=Optimal

Banana = 200.0 grams
GroundBeef = 200.0 grams
LargeEgg = 1000.0 grams
Spinach = 681.8 grams
WhiteRice = 300.0 grams

Daily Cost = $15.05
Weekly Cost = $105.37
```

Now, all 5 foods can be included in the diet (albeit Bananas and Ground Beef are set to the minimum constraint, but for an additional $15 a week, there is significantly more variety in the diet.

Part 5: Utilization of AI models

In this section, Co-Pilot was utilized to attempt to generate a similar Python Script. The entire conversation with Co-Pilot can be found in Conversation with Co-Pilot.txt. This was the following code that was generated:

```
import pulp

# Define the foods and their nutritional values per 100g
foods = ['Ground Beef', 'Spinach', 'White Rice', 'Large Egg', 'Banana']
costs = {'Ground Beef': 1.5, 'Spinach': 0.5, 'White Rice': 0.2, 'Large Egg': 0.3, 'Banana': 0.2}

# Nutritional values per 100g
nutrition = {
    'Ground Beef': {'energy': 250, 'protein': 26, 'vitamin_d': 0, 'calcium': 18, 'iron': 2.6,
'sodium': 72},
    'Spinach': {'energy': 23, 'protein': 2.9, 'vitamin_d': 0, 'calcium': 99, 'iron': 2.7,
'sodium': 79},
    'White Rice': {'energy': 130, 'protein': 2.7, 'vitamin_d': 0, 'calcium': 10, 'iron': 1.2,
'sodium': 1},
    'Large Egg': {'energy': 155, 'protein': 13, 'vitamin_d': 2, 'calcium': 50, 'iron': 1.2,
'sodium': 124},
    'Banana': {'energy': 89, 'protein': 1.1, 'vitamin_d': 0, 'calcium': 5, 'iron': 0.3, 'sodium':
1}
}

# Daily nutritional requirements
requirements = {
    'energy': 2000,
    'protein': 50,
    'vitamin_d': 10,
    'calcium': 1000,
    'iron': 18,
    'sodium': 2300
}

# Define the problem
prob = pulp.LpProblem("Diet Problem", pulp.LpMinimize)

# Define the variables
food_vars = pulp.LpVariable.dicts("Food", foods, lowBound=0, cat='Continuous')

# Objective function: Minimize the cost
prob += pulp.lpSum([costs[i] * food_vars[i] for i in foods]), "Total Cost"

# Constraints
prob += pulp.lpSum([nutrition[i]['energy'] * food_vars[i] for i in foods]) >=
requirements['energy'], "Energy Requirement"
prob += pulp.lpSum([nutrition[i]['protein'] * food_vars[i] for i in foods]) >=
requirements['protein'], "Protein Requirement"
prob += pulp.lpSum([nutrition[i]['vitamin_d'] * food_vars[i] for i in foods]) >=
requirements['vitamin_d'], "Vitamin D Requirement"
prob += pulp.lpSum([nutrition[i]['calcium'] * food_vars[i] for i in foods]) >=
requirements['calcium'], "Calcium Requirement"
prob += pulp.lpSum([nutrition[i]['iron'] * food_vars[i] for i in foods]) >= requirements['iron'],
"Iron Requirement"
prob += pulp.lpSum([nutrition[i]['sodium'] * food_vars[i] for i in foods]) <=
requirements['sodium'], "Sodium Limit"
```

```
# Solve the problem
prob.solve()

# Print the results
print("Status:", pulp.LpStatus[prob.status])
print("Optimal Diet:")
for v in prob.variables():
    if v.varValue > 0:
        print(f"{v.name}: {v.varValue:.2f} grams")
print("Total Cost: ${:.2f}".format(pulp.value(prob.objective)))
```

Without out being provided any nutritional data or constraint data, Co-Pilot was able to generate runnable code that utilizes PuLP correctly. That being said, values determined by the LLM were different than those utilized in the script I wrote. Here was the output of the code above:

```
Status: Optimal
Optimal Diet:
Food_Large_Egg: 12.33 grams
Food_Spinach: 3.87 grams
Total Cost: $5.64
```

The calculated weekly cost was significantly less than what was calculated by my research. It appears that the agent is able to generate the base code needed to solve the linear equation. This confirms that the agent will most likely need value specifications and AI generated values may not be appropriate.