~\Documents\gt2\ISYE6501\HW7\testing.ipynb

```python
import pandas as pd
from pulp import LpProblem, LpVariable, LpMinimize, lpSum, value

# Load the diet data from the specified path
file_path = r'C:\Users\robed\Documents\gt2\ISYE6501\HW7\diet.xls'
df = pd.read_excel(file_path, sheet_name='Sheet1')

# Select the relevant rows of data
diet_data = df.iloc[:64]

# make a list of lists for looping
diet_list = diet_data.values.tolist()

# Create dictionaries for each nutrient and cost based on the diet list
food_items = [item[0] for item in diet_list]
costs = {item[0]: float(item[1]) for item in diet_list}
calories = {item[0]: float(item[3]) for item in diet_list}
cholesterol = {item[0]: float(item[4]) for item in diet_list}
total_fat = {item[0]: float(item[5]) for item in diet_list}
sodium = {item[0]: float(item[6]) for item in diet_list}
carbs = {item[0]: float(item[7]) for item in diet_list}
fiber = {item[0]: float(item[8]) for item in diet_list}
protein = {item[0]: float(item[9]) for item in diet_list}
vitamin_a = {item[0]: float(item[10]) for item in diet_list}
vitamin_c = {item[0]: float(item[11]) for item in diet_list}
calcium = {item[0]: float(item[12]) for item in diet_list}
iron = {item[0]: float(item[13]) for item in diet_list}

# Define the minimum and maximum intake requirements for nutrients
min_nutrients = [1500, 30, 20, 800, 130, 125, 60, 1000, 400, 700, 10]
max_nutrients = [2500, 240, 70, 2000, 450, 250, 100, 10000, 5000, 1500, 40]

# Create a list of nutrient constraints
nutrient_constraints = []
for i in range(11):
    nutrient_constraints.append({item[0]: float(item[i+3]) for item in diet_list})

# Initialize
diet_lp = LpProblem('DietOptimization', LpMinimize)

# Define continuous and binary variables for the food items
food_quantities = LpVariable.dicts("Quantity", food_items, lowBound=0)
is_chosen = LpVariable.dicts("Chosen", food_items, lowBound=0, upBound=1, cat='Binary')

# Define the objective function to minimize the total cost
diet_lp += lpSum([costs[food] * food_quantities[food] for food in food_items]), "Total Cost"

# Add nutrient constraints to the problem
for i in range(11):
    diet_lp += lpSum([nutrient_constraints[i][food] * food_quantities[food] for food in food_items]) >= min_nutrients[i], f"MinNutrient_{i}"
    diet_lp += lpSum([nutrient_constraints[i][food] * food_quantities[food] for food in food_items]) <= max_nutrients[i], f"MaxNutrient_{i}"

```

```
# Solve the optimization problem
diet_lp.solve()

# Print the optimal diet solution
print('Optimal Diet Solution:')
for variable in diet_lp.variables():
    if variable.varValue > 0 and 'Quantity' in variable.name:
        print(f"{variable.varValue:.4f} units of {variable.name.replace('Quantity_', '')}")

# Print the total cost of the optimal diet
print(f"Total cost of food = ${value(diet_lp.objective):.4f}")
```