In [ ]:

```
import pulp

# Instantiate the problem as "Diet Optimization LP"
my_lp_problem = pulp.LpProblem("Diet_Optimization_LP", pulp.LpMinimize)

# Creating Decision Variables for each food item
bar = pulp.LpVariable('bar', lowBound=0, cat='Continuous')
shake = pulp.LpVariable('shake', lowBound=0, cat='Continuous')
paneer = pulp.LpVariable('paneer', lowBound=0, cat='Continuous')
cookies = pulp.LpVariable('cookies', lowBound=0, cat='Continuous')
pasta = pulp.LpVariable('pasta', lowBound=0, cat='Continuous')

# Objective Function
my_lp_problem += 5 * bar + 3.75 * shake + 1.75 * paneer + 2 * cookies + 1.25 * pasta

# Adding Constraints to the problem
my_lp_problem += 130 * bar + 290 * shake + 140 * cookies + 90 * paneer + 190 * pasta >=
my_lp_problem += 130 * bar + 400 * shake + 85 * cookies + 5 * paneer + 0 * pasta <= 5000
my_lp_problem += 12 * bar + 35 * shake + 1 * cookies + 7 * paneer + 4 * pasta >= 800  #
my_lp_problem += 0 * bar + 2.6 * shake + 0 * cookies + 0 * paneer + 0 * pasta >= 20  # V
my_lp_problem += 38 * bar + 300 * shake + 0 * cookies + 125 * paneer + 2 * pasta >= 1300
my_lp_problem += 2 * bar + 5.5 * shake + 1.25 * cookies + 0 * paneer + 0 * pasta >= 18
my_lp_problem += 144 * bar + 318 * shake + 50 * cookies + 0 * paneer + 77 * pasta >= 470

# Solve the LP Problem
my_lp_problem.solve()

# Check the status of the solution
status = pulp.LpStatus[my_lp_problem.status]

if status == 'Optimal':
    # Optimal solution found
    print("Optimal diet:")
    print(f"Servings of Bar: {bar.varValue}")
    print(f"Servings of Shake: {shake.varValue}")
    print(f"Servings of Paneer: {paneer.varValue}")
    print(f"Servings of Cookies: {cookies.varValue}")
    print(f"Servings of Pasta: {pasta.varValue}")
    print(f"Total cost: ${pulp.value(my_lp_problem.objective)}")
else:
    print("No optimal solution found.")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.5
Build Date: May 10 2021

command line - cbc /var/folders/xf/sym_dm2d1dq_7998vxvr369m0000gn/T/0f039b0deb0d486cbc80
009d396b1994-pulp.mps timeMode elapsed branch printingOptions all solution /var/folders/
xf/sym_dm2d1dq_7998vxvr369m0000gn/T/0f039b0deb0d486cbc80009d396b1994-pulp.sol (default s
trategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 12 COLUMNS
At line 44 RHS
At line 52 BOUNDS
At line 53 ENDATA
Problem MODEL has 7 rows, 5 columns and 26 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 3 (-4) rows, 5 (0) columns and 13 (-13) elements
0  Obj 28.846154 Primal inf 22.2524 (2)
3  Obj 143.47255
Optimal - objective value 143.47255
After Postsolve, objective 143.47255, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 143.472552 - 3 iterations time 0.002, Presolve 0.00
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Optimal diet:
Servings of Bar: 0.0
Servings of Shake: 11.900137
Servings of Paneer: 47.989064
Servings of Cookies: 0.0
Servings of Pasta: 11.892942
Total cost: $143.47255325
```

In [ ]:

```python
## I will add two nutritional constraints such as minimum requirements for Vit C and Vit
# #Minimal requirements and nutritional constraints for Vit C are 90 milligrams and Vit

# Instantiate the problem as "Diet Optimization LP"
my_lp_problem = pulp.LpProblem("Diet_Optimization_LP", pulp.LpMinimize)

# Creating Decision Variables for each food item
bar = pulp.LpVariable('bar', lowBound=0, cat='Continuous')
shake = pulp.LpVariable('shake', lowBound=0, cat='Continuous')
paneer = pulp.LpVariable('paneer', lowBound=0, cat='Continuous')
cookies = pulp.LpVariable('cookies', lowBound=0, cat='Continuous')
pasta = pulp.LpVariable('pasta', lowBound=0, cat='Continuous')

# Objective Function
my_lp_problem += 5 * bar + 3.75 * shake + 1.75 * paneer + 2 * cookies + 1.25 * pasta

# Adding Constraints to the problem
my_lp_problem += 130 * bar + 290 * shake + 140 * cookies + 90 * paneer + 190 * pasta >=
my_lp_problem += 130 * bar + 400 * shake + 85 * cookies + 5 * paneer + 0 * pasta <= 5000
my_lp_problem += 12 * bar + 35 * shake + 1 * cookies + 7 * paneer + 4 * pasta >= 800  #
my_lp_problem += 0 * bar + 2.6 * shake + 0 * cookies + 0 * paneer + 0 * pasta >= 20  # V
my_lp_problem += 38 * bar + 300 * shake + 0 * cookies + 125 * paneer + 2 * pasta >= 1300
my_lp_problem += 2 * bar + 5.5 * shake + 1.25 * cookies + 0 * paneer + 0 * pasta >= 18
my_lp_problem += 144 * bar + 318 * shake + 50 * cookies + 0 * paneer + 77 * pasta >= 470
my_lp_problem += 0 * bar + 16 * shake + 0 * cookies + 0 * paneer + 0 * pasta >= 90  # Vi
my_lp_problem += 0 * bar + 0.3 * shake + 0 * cookies + 0 * paneer + 0 * pasta >= 1.7  #

# Solve the LP Problem
```

```
my_lp_problem.solve()

# Check the status of the solution
status = pulp.LpStatus[my_lp_problem.status]

if status == 'Optimal':
    # Optimal solution found
    print("Optimal diet:")
    print(f"Servings of Bar: {bar.varValue}")
    print(f"Servings of Shake: {shake.varValue}")
    print(f"Servings of Paneer: {paneer.varValue}")
    print(f"Servings of Cookies: {cookies.varValue}")
    print(f"Servings of Pasta: {pasta.varValue}")
    print(f"Total cost: ${pulp.value(my_lp_problem.objective)}")
else:
    print("No optimal solution found.")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.5
Build Date: May 10 2021

command line - cbc /var/folders/xf/sym_dm2d1dq_7998vxvr369m0000gn/T/7d638bef3c024401b7f0
cb122133a20c-pulp.mps timeMode elapsed branch printingOptions all solution /var/folders/
xf/sym_dm2d1dq_7998vxvr369m0000gn/T/7d638bef3c024401b7f0cb122133a20c-pulp.sol (default s
trategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 14 COLUMNS
At line 48 RHS
At line 58 BOUNDS
At line 59 ENDATA
Problem MODEL has 9 rows, 5 columns and 28 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 3 (-6) rows, 5 (0) columns and 13 (-15) elements
0  Obj 28.846154 Primal inf 22.2524 (2)
3  Obj 143.47255
Optimal - objective value 143.47255
After Postsolve, objective 143.47255, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 143.472552 - 3 iterations time 0.002, Presolve 0.00
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Optimal diet:
Servings of Bar: 0.0
Servings of Shake: 11.900137
Servings of Paneer: 47.989064
Servings of Cookies: 0.0
Servings of Pasta: 11.892942
Total cost: $143.47255325
```

In [ ]:

In [ ]: