**Syed Fahad Shah**

**BSCS12_C**

**ASSIGNMENT #02 : DESIGN ANALYSIS & ALGORITHM**

# Task 1 A:

## A Political Problem:

Devise a strategy that wins the required number of votes and is also the most inexpensive way of achieving it.

**Variables:**

x1 = number of thousands of dollars spent on advertising on preparing for a zombie apocalypse.

x2 = number of thousands of dollars spent on advertising on equipping sharks with lasers.

x3 = number of thousands of dollars spent on advertising on building highways for buying cars.

x4 = number of thousands of dollars spent on advertising on allowing dolphins to vote.

**Objective Function:**

Z = x1 + x2 + x3 + x4

**Constraints:**

-2*x1 + 8*x2 + 0*x3 + 10*x4 >= 50

5*x1 + 2*x2 + 0*x3 + 0*x4 >= 100

3*x1 – 5*x2 + 10*x3 -2*x4 >= 25

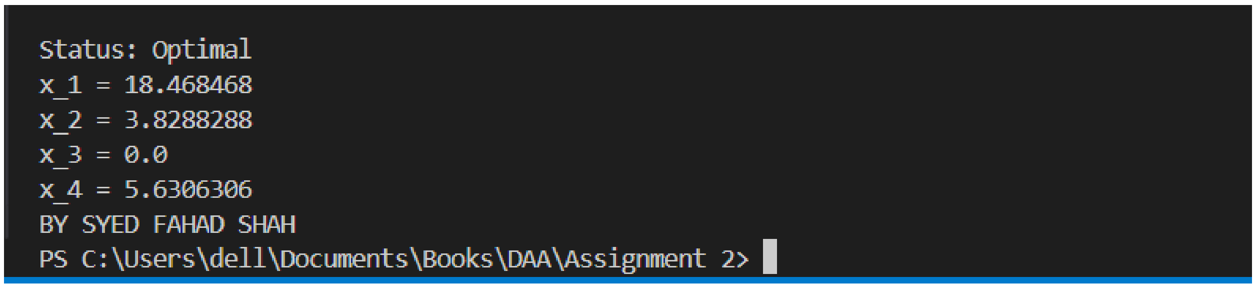
x1, x2, x3, x4 >= 0

**Code:**

```
# SYED FAHAD SHAH
# 407556
#BSCS 12-C
```

```python
#Import Pulp
from pulp import *

#Create a LP Minimization problem
prob = LpProblem("Assignment_2", LpMinimize)

#Create variables
x = LpVariable.dicts("x",[1,2,3,4],lowBound=0,cat='Continuous')

#Objective Function
prob += x[1] + x[2] + x[3] + x[4]

#Constraints

prob += -2*x[1] + 8*x[2] + 0*x[3] + 10*x[4] >= 50
prob += 5*x[1] + 2*x[2] + 0*x[3] + 0*x[4] >= 100
prob += 3*x[1] - 5*x[2] + 10*x[3] - 2*x[4] >= 25

#Solve the problem
status = prob.solve()
print("Status:", LpStatus[status])

#Print the results
for v in prob.variables():
    print(v.name, "=", v.varValue)

print('BY SYED FAHAD SHAH')
```

**OUTPUT:**



```
Status: Optimal
x_1 = 18.468468
x_2 = 3.8288288
x_3 = 0.0
x_4 = 5.6306306
BY SYED FAHAD SHAH
PS C:\Users\dell\Documents\Books\DAA\Assignment 2>
```

## Task 1 B:

### Wyndor Glass Problem:

Determine what mixture of the two products would result in higher profits for the company.

**Variables:**

x1 = Glass 1

x2 = Glass 2

**Objective Function:**

Z = 3*x1 + 5*x2

**Constraints:**

x1 <= 4

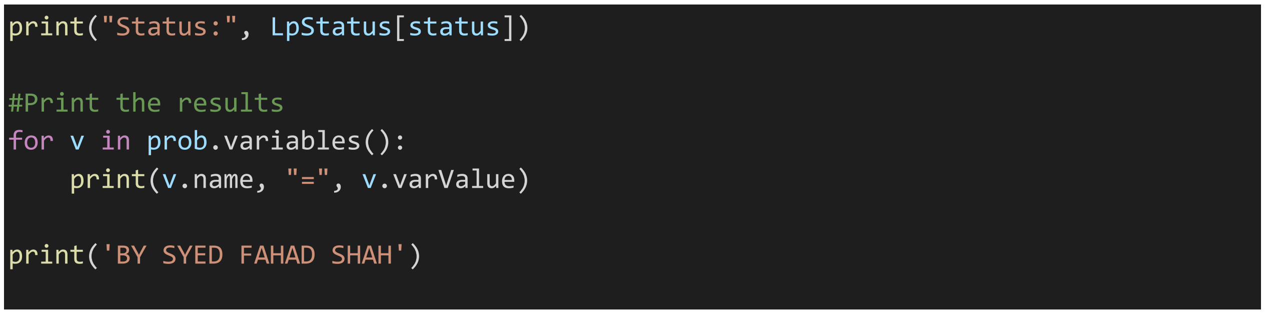
2*x2 <= 12

3*x1 + 2*x2 <= 18

x1, x2 >= 0

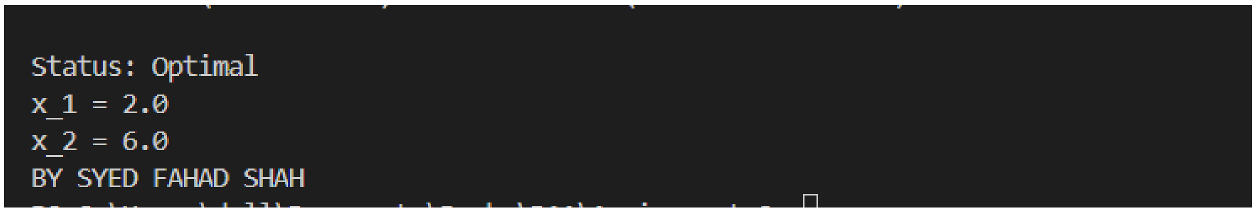
**Code:**

```python
# SYED FAHAD SHAH
# 407556
#BSCS 12-C
#Import Pulp
from pulp import *

#Create a LP Maximization problem
prob = LpProblem("Assignment_2", LpMaximize)

#Create variables
x = LpVariable.dicts("x",[1,2],lowBound=0,cat='Continuous')

#Objective Function
prob += 3*x[1] + 5*x[2]

#Constraints
prob += x[1] <= 4
prob += 2*x[2] <= 12
prob += 3*x[1] + 2*x[2] <= 18

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



```python
print("Status:", LpStatus[status])

#Print the results
for v in prob.variables():
    print(v.name, "=", v.varValue)

print('BY SYED FAHAD SHAH')
```

OUTPUT:



```
Status: Optimal
x_1 = 2.0
x_2 = 6.0
BY SYED FAHAD SHAH
```

## TASK 1 C:

## Custom LP Problem

Problem Statement:

A company produces two products, Product A and Product B. Each product requires a certain amount of labor and materials to produce. The company wants to maximize its profit while ensuring that it meets labor and material availability constraints.

Linear Programming Model:

Let:

- x1: Quantity of Product A produced (in units)
- x2: Quantity of Product B produced (in units)

**Objective function:** Maximize 10(x1)+15(x2)  (representing profit)

**Subject to:**

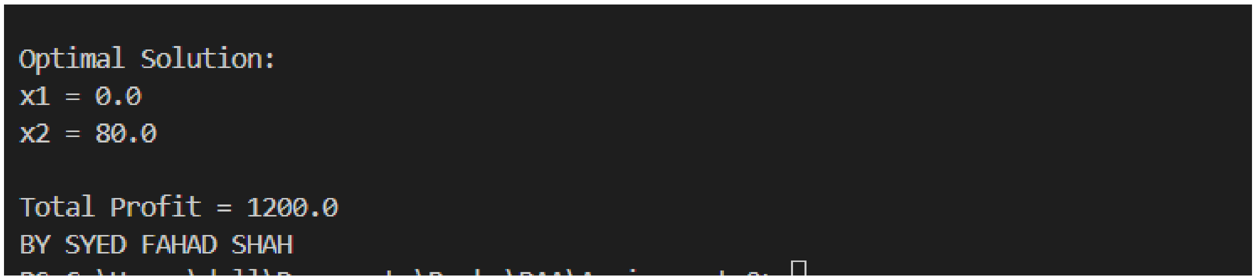
- 2(x1) +3(x2) ≤ 240 (labor availability constraint)
- 4(x1) + 2(*x*2) ≤400 (material availability constraint)

All variables x1, x2  ≥ 0

**PuLP Code:**

```python
#SYED FAHAD SHAh
#BSCS12-C
#CMS:  407556
import pulp

# Create a LP maximization problem
prob = pulp.LpProblem("Custom_LP_Problem", pulp.LpMaximize)

# Define decision variables
x1 = pulp.LpVariable('x1', lowBound=0, cat='Continuous')
x2 = pulp.LpVariable('x2', lowBound=0, cat='Continuous')

# Objective function
prob += 10*x1 + 15*x2, "Total Profit"

# Constraints
prob += 2*x1 + 3*x2 <= 240
prob += 4*x1 + 2*x2 <= 400

# Solve the problem
prob.solve()

# Print the results
print("Optimal Solution:")
for v in prob.variables():
    print(v.name, "=", v.varValue)

print("\nTotal Profit =", pulp.value(prob.objective))
print('BY SYED FAHAD SHAH')
```

**OUTPUT:**



```
Optimal Solution:
x1 = 0.0
x2 = 80.0

Total Profit = 1200.0
BY SYED FAHAD SHAH
```

## TASK 2:

In the political problem at the beginning of this chapter, there are feasible solutions that correspond to winning more voters than there actually are in the district. For example, you can set x2 to 200, x3 to 200, and x1 D x4 D 0. That solution is feasible, yet it seems to say that you will win 400,000 suburban voters, even though there are only 200,000 actual suburban voters. What constraints can you add to the linear program to ensure that you never seem to win more voters than there actually are? Even if you don't add these constraints, argue that the optimal solution to this linear program can never win more voters than there actually are in the district.

**SOLUTION:**

To ensure that the solution to the political problem does not indicate winning more voters than there actually are, we can add constraints to limit the total number of voters won by the candidate in each demographic group.

Let's denote the number of voters in each demographic group as follows:

V1: Urban voters

V2: Suburban voters

V3: Rural voters

We need to ensure that the total number of voters won in each demographic group does not exceed the actual number of voters in that group. Mathematically, we can express this as:

$$X1+ X4 <= V1$$

$$X2 + X5 <= V2$$

$$X3 + X6 <= V3$$

Where X4, X5, and X6 represent the support gained in each demographic group through channels other than television, radio, and print ads respectively.

Adding these constraints ensures that the optimal solution does not indicate winning more voters than actually exist in each demographic group.

Additionally, we can argue that the optimal solution to the linear program can never win more voters than there actually are in the district based on the nature of linear programming and the constraints provided in the problem. The constraints in the problem are derived from real-world limitations such as budget constraints and limits on advertising channels. These constraints inherently restrict the amount of support that can be gained in each demographic group. Therefore, the optimal solution, which maximizes support while satisfying these constraints, cannot result in winning more voters than actually exist in the district.