Rabin Gurung
Project 5

- **The Statement of the problem**
  A car company uses robots to manufacture cars. The following demands for cars must be met (not necessarily on time, but all demands must be met by the end of month 4).

Month	Demand
1	600
2	800
3	500
4	400

At the beginning of the month, company has two robots. Robots can be purchased at the beginning of each month, but a maximum of two per month can be purchased. Each robot can build at most 200 cars per month. It costs $5,000 to purchase a robot. Each month, a robot incurs $500 in maintenance costs (even if it is not used to build any cars). Robots can also be sold at the beginning of each month for $3,000. At the end of each month, a holding cost of $200 per car is incurred. If any demand is backlogged, then a cost of $400 per car is incurred for each month the demand is backlogged. At the end of month 4, the company must have at least two robots. Formulating an LP to minimize the total cost incurred in meeting the next four months' demands for cars.

- **Decision Variables:**

Rn: robots available during month n (after robots are bought or sold for the month)

Bn: robots bought during month n

Sn: robots sold during month n

In: cars in inventory at end of month n

Cn: cars produced during month n

Dn: backlogged demand for cars at end of month n

Where, n = {1 - 4} which go from month 1, month 2, month 3, month4

- **LP formulation:**

Objective function:

Rabin Gurung
Project 5

$MIN\ 500(R1 + R2 + R3 + R4) + 200(I1 + I2 + I3 + I4) + 5000(B1 + B2 + B3 + B4) - 3000(S1 + S2 + S3 + S4) + 300(D1 + D2 + D3 + D4)$

S.T.
Constraints:

$R1 = 2 + B1 - S1$

$R2 = R1 + B2 - S2$

$R3 = R2 + B3 - S3$

$R4 = R3 + B4 - S4$

$I1 - D1 = C1 - 600$

$I2 - D2 = I1 - D1 + C2 - 800$

$I3 - D3 = I2 - D2 + C3 - 500$

$I4 - D4 = I3 - D3 + C4 - 400$

$R4 >= 2$

$C1 <= 200 R1$

$C2 <= 200 R2$

$C3 <= 200 R3$

$C4 <= 200 R4$

$D4 = 0$

$B1 <= 2$

$B2 <= 2$

$B3 <= 2$

$B4 <= 2$

Rabin Gurung
Project 5

$\{R1, R2, R3, R4, B1, B2, B3, B4, S1, S2, S3, S4, I1, I2, I3, I4, C1, C2, C3, C4, D1, D2, D3, D4\} >= 0$

Some of the equalities are converted to inequalities to maintain the standard form, so I can use the constraints in the python program.

Objective Function

$500 * (R1 + R2 + R3 + R4) + 200 * (I1 + I2 + I3 + I4) + 5000 * (B1 + B2 + B3 + B4) - 3000 * (S1 + S2 + S3 + S4) + 300 * (D1 + D2 + D3 + D4)$

Constraints:

$R1 - B1 + S1 >= 2$
$- R1 + B1 - S1 >= -2$

$R2 - R1 - B2 + S2 >= 0$
$- R2 + R1 + B2 - S2 >= 0$

$R3 - R2 - B3 + S3 >= 0$
$-R3 + R2 + B3 - S3 >= 0$

$R4 - R3 - B4 + S4 >= 0$
$-R4 + R3 + B4 - S4 >= 0$

$I1 - D1 - C1 >= 600$
$-I1 + D1 + C1 >= -600$

$I2 - D2 - I1 + D1 - C2 >= 800$
$-I2 + D2 + I1 - D1 + C2 >= -800$

$I3 - D3 - I2 + D2 - C3 >= 500$
$-I3 + D3 + I2 - D2 + C3 >= -500$

$I4 - D4 - I3 + D3 - C4 >= 400$
$-I4 + D4 + I3 - D3 + C4 >= -400$

$R4 >= 2$

$C1 - 200 * R1 <= 0$
$C2 - 200 * R2 <= 0$
$C3 - 200 * R3 <= 0$
$C4 - 200 * R4 <= 0$

$D4 >= 0$

-D4 >= 0

B1 <= 2
B2 <= 2
B3 <= 2
B4 <= 2

{R1, R2, R3, R4, I1, I2, I3, I4, B1, B2, B3, B4, S1, S2, S3, S4, C1, C2, C3, C4, D1, D2, D3, D4} >= 0

Explanation:
Steps are explained below.

Line 1-2: First import the library pulp as p.
Line 4-5: Define the problem by giving a suitable name to your problem, here I have given the name 'Problem'. Also, specify your aim for the objective function of whether to Maximize or Minimize.
Line 7-31: Define LpVariable to hold the variables of the objective functions. The next argument specifies the lower bound of the defined variable, i.e. 0, and the upper bound is none by default. You can specify the upper bound too.
Line 33-34: Denotes the objective function in terms of defined variables.
Line 36-74: These are the constraints on the variables.
Line 76- 77: This will show you the problem in the output screen.
Line 79: This is the problem solver.
Line 80: Will display the status of the problem.
Line 83: Will print the value for R1, R2, R3, R4, I1, I2, I3, I4, B1, B2, B3, B4, S1, S2, S3, S4, C1, C2, C3, C4, D1, D2, D3, and D4 and the minimum value for the objective function.

Final Result given by the computer program,

Rabin Gurung
Project 5

```
Problem:
MINIMIZE
5000*B1 + 5000*B2 + 5000*B3 + 5000*B4 + 300*D1 + 300*D2 + 300*D3 + 300*D4 + 200*I1 + 200*I2 + 200*I3 + 200*I4 + 500*R1 + 500*R2 + 500*R3 + 500*R4 + -3000
*S1 + -3000*S2 + -3000*S3 + -3000*S4 + 0
SUBJECT TO
_C1: - B1 + R1 + S1 >= 2

_C2: B1 - R1 - S1 >= -2

_C3: - B2 - R1 + R2 + S2 >= 0

_C4: B2 + R1 - R2 - S2 >= 0

_C5: - B3 - R2 + R3 + S3 >= 0

_C6: B3 + R2 - R3 - S3 >= 0

_C7: - B4 - R3 + R4 + S4 >= 0

_C8: B4 + R3 - R4 - S4 >= 0

_C9: - C1 - D1 + I1 >= 600

_C10: C1 + D1 - I1 >= -600

_C11: - C2 + D1 - D2 - I1 + I2 >= 800

_C12: C2 - D1 + D2 + I1 - I2 >= -800

_C13: - C3 + D2 - D3 - I2 + I3 >= 500

_C14: C3 - D2 + D3 + I2 - I3 >= -500

_C15: - C4 + D3 - D4 - I3 + I4 >= 400

_C16: C4 - D3 + D4 + I3 - I4 >= -400

_C17: R4 >= 2

_C18: C1 - 200 R1 <= 0
```

```
_C19: C2 - 200 R2 <= 0

_C20: C3 - 200 R3 <= 0

_C21: C4 - 200 R4 <= 0

_C22: D4 >= 0

_C23: - D4 >= 0

_C24: B1 <= 2

_C25: B2 <= 2

_C26: B3 <= 2

_C27: B4 <= 2

VARIABLES
B1 Continuous
B2 Continuous
B3 Continuous
B4 Continuous
C1 Continuous
C2 Continuous
C3 Continuous
C4 Continuous
D1 Continuous
D2 Continuous
D3 Continuous
D4 Continuous
I1 Continuous
I2 Continuous
I3 Continuous
I4 Continuous
R1 Continuous
R2 Continuous
R3 Continuous
R4 Continuous
S1 Continuous
```

Rabin Gurung
Project 5

```
R3 Continuous
R4 Continuous
S1 Continuous
S2 Continuous
S3 Continuous
S4 Continuous

Welcome to the CBC MILP Solver
Version: 2.9.0
Build Date: Feb 12 2015

command line - /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/pulp/apis/../solverdir/cbc/osx/64/cbc /var/folders/gz/fkdnzf
g15pn4tpmvtw7l7p6h0000gn/T/6f341bab1d994e2c91f0dfae453ddb8d-pulp.mps ratio None allow None threads None presolve on strong None gomory on knapsack on pro
bing on branch printingOptions all solution /var/folders/gz/fkdnzfg15pn4tpmvtw7l7p6h0000gn/T/6f341bab1d994e2c91f0dfae453ddb8d-pulp.sol (default strategy
1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 32 COLUMNS
At line 134 RHS
At line 162 BOUNDS
At line 163 ENDATA
Problem MODEL has 27 rows, 24 columns and 81 elements
Coin0008I MODEL read with 0 errors
String of None is illegal for double parameter ratioGap value remains 0
String of None is illegal for double parameter allowableGap value remains 0
String of None is illegal for integer parameter threads value remains 0
String of None is illegal for integer parameter strongBranching value remains 5
Option for gomoryCuts changed from ifmove to on
Option for knapsackCuts changed from ifmove to on
Presolve 20 (-7) rows, 23 (-1) columns and 72 (-9) elements
0  Obj 120980 Primal inf 1197.6747 (7) Dual inf 6188.0312 (4)
12  Obj 1244000
Optimal - objective value 1244000
After Postsolve, objective 1244000, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 1244000 - 12 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
2.0 2.0 2.0 2.0 600.0 1400.0 1900.0 2300.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1244000.0
```

This result means optimal value for Robots available after robots are bought or sold for the month 1- 4 are R1 = 2, R2 = 2, R3 = 2, R4 = 2. Optimal value of inventory of the Robots are from month 1 - 4 are 600, 1400, 1900, 2300 respectively. Optimal value of rest of other variables are zero. The optimized objective function value is 1244000 which means minimum optimized cost to meet the demands of cars produced in these four months.