In [1]:
```
from pulp import LpMaximize, LpProblem, LpStatus, LpVariable, LpMinimize,
```

In [2]:
```
# Set up minimization model
supply_chain_model = LpProblem(name='supply_chain', sense=LpMinimize)
supply_chain_model
```

Out[2]:
```
supply_chain:
MINIMIZE
None
VARIABLES
```

In [3]:
```
# Define the decision variables
A_1_1 = LpVariable(name="A_1_1", lowBound=0, upBound=3000, cat="Integer")
A_1_3 = 3000 - A_1_1
A_1_4 = LpVariable(name="A_1_4", lowBound=0, upBound=2000, cat="Integer")
A_1_5 = 2000 - A_1_4
B_1_1 = LpVariable(name="B_1_1", lowBound=0, upBound=1000, cat="Integer")
B_1_2 = LpVariable(name="B_1_2", lowBound=0, upBound=1000, cat="Integer")
B_1_3 = (1000 - B_1_1) + (1000 - B_1_2)
B_1_4 = LpVariable(name="B_1_4", lowBound=0, upBound=2000, cat="Integer")
B_1_5 = 2000 - B_1_4
C_2_1 = LpVariable(name="C_2_1", lowBound=0, upBound=2000, cat="Integer")
C_2_2 = LpVariable(name="C_2_2", lowBound=0, upBound=2000, cat="Integer")
C_2_3 = (2000 - C_2_1) + (2000 - C_2_2)
D_2_2 = LpVariable(name="D_2_2", lowBound=0, upBound=1000, cat="Integer")
D_2_3 = 1000 - D_2_2
D_2_4 = LpVariable(name="D_2_4", lowBound=0, upBound=3000, cat="Integer")
D_2_5 = 3000 - D_2_4
E_3_1 = LpVariable(name="E_3_1", lowBound=0, upBound=1000, cat="Integer")
E_3_2 = LpVariable(name="E_3_2", lowBound=0, upBound=1000, cat="Integer")
E_3_3 = (1000 - E_3_1) + (1000 - E_3_2)
E_3_4 = LpVariable(name="E_3_4", lowBound=0, upBound=2000, cat="Integer")
E_3_5 = 2000 - E_3_4
```

In [4]:

```python
# Define Objective Function
obj_func = (
    A_1_1 * 2
    + A_1_3 * 4
    + A_1_4 * 6
    + A_1_5 * 8
    + B_1_1 * 2
    + B_1_2 * 3
    + B_1_3 * 4
    + B_1_4 * 6
    + B_1_5 * 8
    + C_2_1 * 4
    + C_2_2 * 2
    + C_2_3 * 5
    + D_2_2 * 2
    + D_2_3 * 5
    + D_2_4 * 7
    + D_2_5 * 9
    + E_3_1 * 3
    + E_3_2 * 4
    + E_3_3 * 2
    + E_3_4 * 5
    + E_3_5 * 7
) * 1000

# Add Objective function to the model
supply_chain_model += obj_func
supply_chain_model
```

Out[4]:

```
supply_chain:
MINIMIZE
-2000*A_1_1 + -2000*A_1_4 + -2000*B_1_1 + -1000*B_1_2 + -2000*B_1_4 + -1000
*C_2_1 + -3000*C_2_2 + -3000*D_2_2 + -2000*D_2_4 + 1000*E_3_1 + 2000*E_3_2
+ -2000*E_3_4 + 122000000
VARIABLES
0 <= A_1_1 <= 3000 Integer
0 <= A_1_4 <= 2000 Integer
0 <= B_1_1 <= 1000 Integer
0 <= B_1_2 <= 1000 Integer
0 <= B_1_4 <= 2000 Integer
0 <= C_2_1 <= 2000 Integer
0 <= C_2_2 <= 2000 Integer
0 <= D_2_2 <= 1000 Integer
0 <= D_2_4 <= 3000 Integer
0 <= E_3_1 <= 1000 Integer
0 <= E_3_2 <= 1000 Integer
0 <= E_3_4 <= 2000 Integer
```

In [5]:

```
status = supply_chain_model.solve()
print ("status returned code :", status)
print ({0: 'Not Solved', 1: 'Optimal', -1: 'Infeasible', -2: 'Unbounded',
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/johua/.pyenv/versions/3.9.0/lib/python3.9/site-packag
es/pulp/solverdir/cbc/osx/64/cbc /var/folders/ff/nyt4g8zd03d1s5qfny3sg36c00
00gn/T/d9b0e819f5514be0871c2f10515c442d-pulp.mps timeMode elapsed branch pr
intingOptions all solution /var/folders/ff/nyt4g8zd03d1s5qfny3sg36c0000gn/
T/d9b0e819f5514be0871c2f10515c442d-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 5 COLUMNS
At line 42 RHS
At line 43 BOUNDS
At line 56 ENDATA
Problem MODEL has 0 rows, 12 columns and 0 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is -3.8e+07 - 0.00 seconds
Cgl0004I processed model has 0 rows, 0 columns (0 integer (0 of which binar
y)) and 0 elements
Cbc3007W No integer variables - nothing to do
Cuts at root node changed objective from -3.8e+07 to -1.79769e+308
Probing was tried 0 times and created 0 cuts of which 0 were active after a
dding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after ad
ding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were
active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active afte
r adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after
adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:                -38000000.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.01

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.01

status returned code : 1
{0: 'Not Solved', 1: 'Optimal', -1: 'Infeasible', -2: 'Unbounded', -3: 'Und
efined'}
```

In [6]:

```
print("Solution is", LpStatus[supply_chain_model.status])
print("objective :", value(supply_chain_model.objective))

for var in supply_chain_model.variables():
    print(var.name, ":", var.value())
for name, constraint in supply_chain_model.constraints.items():
    print(name, " : ", constraint.value())
```

```
Solution is Optimal
objective : 84000000.0
A_1_1 : 3000.0
A_1_4 : 2000.0
B_1_1 : 1000.0
B_1_2 : 1000.0
B_1_4 : 2000.0
C_2_1 : 2000.0
C_2_2 : 2000.0
D_2_2 : 1000.0
D_2_4 : 3000.0
E_3_1 : 0.0
E_3_2 : 0.0
E_3_4 : 2000.0
```

Optimal shipping plan

(values are given in thousands of units)

- Product A
  - Produce at facility 1 and send to distribution center 1: 3000
    - Region 1: 3000
  - Produce at facility 1 and send to distribution center 3: 0
  - Produce at facility 1 and send to distribution center 4: 2000
    - Region 3: 2000
  - Produce at facility 1 and send to distribution center 5: 0
- Product B
  - Produce at facility 1 and send to distribution center 1: 1000
    - Region 1: 1000
  - Produce at facility 1 and send to distribution center 2: 1000
    - Region 2: 1000
  - Produce at facility 1 and send to distribution center 3: 0
  - Produce at facility 1 and send to distribution center 4: 2000
    - Region 3: 1000
    - Region 4: 1000
  - Produce at facility 1 and send to distribution center 5: 0
- Product C
  - Produce at facility 2 and send to distribution center 1: 2000
    - Region 1: 2000
  - Produce at facility 2 and send to distribution center 2: 2000
    - Region 2: 2000
  - Produce at facility 2 and send to distribution center 3: 0
- Product D
  - Produce at facility 2 and send to distribution center 2: 1000
    - Region 2: 1000
  - Produce at facility 2 and send to distribution center 3: 0
  - Produce at facility 2 and send to distribution center 4: 3000
    - Region 4: 3000
  - Produce at facility 2 and send to distribution center 5: 0

- Product E
  - Produce at facility 3 and send to distribution center 1: 0
  - Produce at facility 3 and send to distribution center 2: 0
  - Produce at facility 3 and send to distribution center 3: 2000
    - Region 1: 1000
    - Region 2: 1000
  - Produce at facility 3 and send to distribution center 4: 2000
    - Region 3: 1000
    - Region 4: 1000
  - Produce at facility 3 and send to distribution center 5: 0