For this assignment I used Python and PuLP package to solve a problem found on the Purple Math website. The following calculators' production problem was taken from a list of word problems at: https://www.purplemath.com/modules/linprog3.htm

Problem:

A calculator company produces a scientific calculator and a graphing calculator. Long-term projections indicate an expected demand of at least 100 scientific and 80 graphing calculators each day. Because of limitations on production capacity, no more than 200 scientific and 170 graphing calculators can be made daily. To satisfy a shipping contract, a total of at least 200 calculators much be shipped each day.

If each scientific calculator sold results in a $2 loss, but each graphing calculator produces a $5 profit, how many of each type should be made daily to maximize net profits?

Scenario 1:

First, I imported the pulp package and created a variable to hold the problem data using the following code:

```
#imports
from pulp import * #LP modeller

#Create a variable to contain the problem data
sales=LpProblem("CalculatorSales", LpMaximize)
```

Then, I created two variables for the scientific and graphic calculators to be produced. Since LpVariable() function allows us to specify lower and upper bounds for the variables directly, I used variable bounds instead of adding expected minimum demand and maximum production capabilities for each of the two types of the calculators as the model constraints.

I used the following code:

```
#Create variables for scientific and graphing calculators
#Based on the projected demand
#lower limits for 100 for scientific calculators and 80 for graphing calculators
#Based on production limitations upper limits 200 for scientific and 170 for graphing
calculators
i=LpVariable("scientific", 100, 200, LpInteger)
j=LpVariable("graphing", 80, 170, LpInteger)
```

The goal, or objective function, is to maximize profits from manufacturing and selling the calculators. If the company looses $2 on each scientific calculator and earns $5 on each graphing calculator, than total sales amount that we need to maximize can be expressed using the following function:

```
# Add the objective function - maximise profits
sales += (-2)*i + 5 * j, "total $ sales"
```

The only constraint in this model is the contractual obligation to ship at least 200 calculators each day, and it can be expressed using the following code:

```
#Add min shipment constraint constraint
sales += i + j >= 200, "min daily shipment" #contractual obligation to ship at least
200 calculators per day
```

The next step is to write data in to an .lp file and use the pulp solver to find a solution:

```
#write data to an .lp file
sales.writeLP("Calculators.lp")

#Solve the problem using PuLP solver
sales.solve()
```

To view the status of the solution I used the following code:

```
#View the status of the solution
print('Scenario 1:')
print(LpStatus[sales.status])
```

Which returned that we have an optimal solution:

```
Scenario 1:

Optimal
```

I used the following code to view the variables (number of each of the two type of calculators to manufacture) and the optimum objective function value (amount of sales):

```
#View each variable for the optimum solution
for x in sales.variables():
    print(x.name, "=", x.varValue)

#View the optimized objective function
print('Total Calculators Sales $', value(sales.objective))
```

Which returned the following:

```
graphing = 170.0

scientific = 100.0

Total Calculators Sales $ 650.0
```

Conclusion:

The optimum solution to satisfy minimum manufacturing and shipping requirements while maximizing profit would be to produce 170 graphing calculators and 100.00 scientific calculators daily. It would bring $650.00 in profits.

Scenario 2:

Then I decided to modify conditions of the initial problem a little assuming that instead of $2 loss each scientific calculator would bring $1 profit. All other parameters remained the same. I used a similar procedure:

```
################Modify the problem: Scenario 2################################
#########Scientific calculators bring $1 of profit (instead of $2 loss)#######
#########All other conditions remain the same#################################
##############################################################################


#Create a variable to contain the problem data
sales2=LpProblem("CalculatorSales@", LpMaximize)

#Create variables for scientific and graphing calculators
#Based on the projected demand
#lower limits for 100 for scientific calculators and 80 for graphing calculators
#Based on production limitations upper limits 200 for scientific and 170 for graphing
calculators
i2=LpVariable("scientific", 100, 200, LpInteger)
j2=LpVariable("graphing", 80, 170, LpInteger)


# Add the objective function - maximize profits
sales2 += i2 + 5 * j2, "total $ sales"

#Add min shipment constraint
sales2 += i2 + j2 >= 200, "min daily shipment" #contractual obligation to ship at
least 200 calculators per day

#write data to an .lp file
sales2.writeLP("Calculators.lp")

#Solve the problem using PuLP solver
sales2.solve()

print('Scenario 2:')

#View the status of the solution
print(LpStatus[sales2.status])

#View each variable for the optimum solution
for z in sales2.variables():
    print(z.name, "=", z.varValue)

#View the optimized objective function
print('Total Calculators Sales $', value(sales2.objective))
```

With the following results:

```
Scenario 2:

Optimal

graphing = 170.0
```

scientific = 200.0

Total Calculators Sales $1050.0

Conclusion:

Maximum potential profit of $1050.00 would be received if we use our manufacturing facility at its production capacity – 170.00 graphing and 200.00 scientific calculators.

Scenario 3:

In the third scenario I looked at the optimum production in a situation in which each scientific calculator produces $0 results (neither profit, no loss), all other constraints being the same.

As in the previous scenarios, I used the following code:

```
################Modify the problem: Scenario 3################################
#########Scientific calculators bring $0 of profit (instead of $2 loss)#######
#########All other conditions remain the same################################
############################################################################


#Create a variable to contain the problem data
sales3=LpProblem("CalculatorSales@", LpMaximize)

#Create variables for scientific and graphing calculators
#Based on the projected demand
#lower limits for 100 for scientific calculators and 80 for graphing calculators
#Based on production limitations upper limits 200 for scientific and 170 for graphing
calculators
i3=LpVariable("scientific", 100, 200, LpInteger)
j3=LpVariable("graphing", 80, 170, LpInteger)


# Add the objective function - maximize profits
sales3 += 5 * j3, "total $ sales"

#Add min shipment constraint
sales3 += i3 + j3 >= 200, "min daily shipment" #contractual obligation to ship at
least 200 calculators per day

#write data to an .lp file
sales3.writeLP("Calculators.lp")

#Solve the problem using PuLP solver
sales3.solve()

print('Scenario 3:')

#View the status of the solution
print(LpStatus[sales3.status])

#View each variable for the optimum solution
for y in sales3.variables():
    print(y.name, "=", y.varValue)

#View the optimized objective function
print('Total Calculators Sales $', value(sales3.objective))
```

The PuLP modeler returned the following results:

```
Scenario 3:

Optimal

graphing = 170.0

scientific = 200.0

Total Calculators Sales $ 850.0
```

Conclusion:

Even if the scientific calculators bring either profit or loss, the model recommends to manufacture calculators at the maximum capacity – 170.00 graphing and 200.00 scientific.