LINEAR PROGRAMMING

SIMPLEX METHOD

CODING ASSIGNMENT

Solutions to Linear Programming problems can be obtained by using the PuLP Python package

Instructions for installing pulp package

http://pythonhosted.org/PuLP/main/installing_pulp_at_home.html

In [17]:
```
!pip install pulp
```

```
Requirement already satisfied: pulp in c:\users\rohit\anaconda3\lib\site-packages (2.
7.0)
```

Problem 1: Landon runs a bakery that sells two kinds of pies. Landon knows the bakery must make at least 2 and at most 78 dozens of the Lemon Puckers. The bakery must also make at least 2 and at most 42 dozens of the Mint Breezes. Each tray of Lemon Puckers takes 14 ounces of flour, while each tray of Mint Breezes requires 13 ounces of flour. The bakery only has 1274 ounces of flour available. If dozens of Lemon Puckers generate $2.73 in income, and dozens of Mint Breezes generate $1.66, how many dozens of the pies should Landon have the bakery make to get the most income?

Desired Output

![image.png](attachment:image.png)

Solution to Problem 1

In [14]:
```
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize

# Declaring variables

LemonPuckers = LpVariable("LemonPucker", lowBound = 2,upBound=78)
MintBreezes = LpVariable("MintBreezes", lowBound = 2, upBound=42)

# Defining the problem

problem = LpProblem("problem",LpMaximize)

# Defining the constraints

problem += (14*LemonPuckers) + (13 * MintBreezes) <= 1274
```

```
# Defining the objective constraint
problem += (2.73*LemonPuckers) + (1.66*MintBreezes)

# Solving the problem
solution = problem.solve()
LpStatus[solution]

print(f"Number (in dozens) of Lemon Puckers = {int(value(LemonPuckers))}, Mint Breezes
print(f"Maximum profit is: ${(2.73*value(LemonPuckers)) + (1.66*value(MintBreezes))}")
```

```
Number (in dozens) of Lemon Puckers = 78, Mint Breezes = 14
Maximum profit is: $236.18
```

Problem 2: A company has $10,550 available per month for advertising. Newspaper ads cost $190 each and can't run more than 25 times per month. Radio ads cost $500 each and can't run more than 32 times per month at this price. Each newspaper ad reaches 7000 potential customers, and each radio ad reaches 8400 potential customers. The company wants to maximize the number of ad exposures to potential customers.

- Use n for number of Newspaper advertisements and r for number of Radio advertisements.
- Round ads and group exposure to the nearest whole person.

Desired Output

`![image-2.png](attachment:image-2.png)`

Solution to Problem 2

In [7]:

```
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize

# Declaring variables
n = LpVariable("NewspaperAds", upBound = 25) # Newspaper ads
r = LpVariable("RadioAds", upBound = 32) # use Radio Advertisements

# Defining the problem
problem = LpProblem("problem",LpMaximize)

# Defining constraints
problem += (190 * n) + (500*r) <= 10550

#Defining objective function

problem +=  (7000 * n) + (8400 * r)

#Solving the problem
solution = problem.solve()
LpStatus[solution]

print(f"Number of newspaper ads = {int(value(n))}, radio ads = {int(value(r))}")
print(f"Maximum exposure is {int((7000 * value(n)) + (8400 * value(r)))}")
```

```
Number of newspaper ads = 25, radio ads = 11
Maximum exposure is 272440
```

Problem 3: A factory manufactures three products, A, B, and C. Each product requires the use of two machines, Machine I and Machine II. The total hours available, respectively, on Machine I and Machine II per month are 8,070 and 10,380. The time requirements and profit per unit for each product are listed below.

`![image.png](attachment:image.png)`

How many units of each product should be manufactured to maximize profit, and what is the maximum profit? Round numbers of items to the nearest whole number and profit to 2 decimal places.

Desired Output

`![image-2.png](attachment:image-2.png)`

Solution to Problem 3

In [14]:
```
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize

# Declaring variables
a = LpVariable("ProductA",lowBound = 0)
b = LpVariable("ProductB",lowBound = 0)
c = LpVariable("ProductC",lowBound=0)

# Defining Problem
problem = LpProblem("problem", LpMaximize)

# Defining Constraints
problem += (7*a) + (10*b) + (10*c) <= 8070
problem += (10*a) + (9*b) + (16*c) <= 10380

# Defining objective function
problem += (8*a) + (14*b) + (19*c)

# Solving the problem
solution = problem.solve()
LpStatus[solution]

print(f"Units of Product A = {int(value(a))}, Units of Product B = {int(value(b))}, Ur
print(f"Maximum profits are ${((8*value(a)) + (14*value(b)) + (19*value(c))):.2f}")
```

```
Units of Product A = 0, Units of Product B = 361, Units of Product C = 445
Maximum profits are $13524.43
```

Problem 4: The water-supply manager for Cincinnati needs to supply the city with at least 20 million gallons of potable water per day. The supply may be drawn from the local reservoir or from a pipeline to an adjacent town. The local reservoir has a maximum daily yield of 20 million gallons of potable water, and the pipeline has a maximum daily yield of 14 million gallons. By contract, the pipeline is required to supply a minimum of 13 million

gallons per day. If the cost for 1 million gallons of reservoir water is $310 and the cost for 1 million gallons of pipeline water is $290, how much water should the manager get from each source to minimize daily water costs for the city? What is the minimum daily water cost?
- All answers should be integer values.

Desired Output

`![image.png](attachment:image.png)`

Solution to Problem 4

In [18]:
```
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize

# Declaring variables
localres = LpVariable("localres", upBound = 20)
pipeline = LpVariable("pipeline", lowBound = 13, upBound = 14)

#Defining the problem
problem = LpProblem("problem", LpMinimize)

#Defining the constraints
problem += (pipeline + localres) >= 20

#Defining the objective function
problem += (310*localres) + (290*pipeline)

#Solving the problem
solution = problem.solve()
LpStatus[solution]

print(f"Water from each source (in million gallons): local reservoir = {int(value(loca
print(f"Minimum cost is {int((310*value(localres)) + (290*value(pipeline)))}")
```

```
Water from each source (in million gallons): local reservoir = 6, pipeline = 14
Minimum cost is 5920
```

Problem 5: A diet is to contain at least 2950 mg vitamin C, 2890 mg Calcium, and 2430 calories every day. Two foods, a dairy-based meal and a vegan option are to fulfill these requirements. Each ounce of the dairy-based meal provides 50 mg vitamin C, 30 mg Calcium, and 10 calories. Each ounce of the vegan option provides 20 mg vitamin C, 40 mg Calcium, and 50 calories. If the dairy-based meal costs $0.33 per ounce and the vegan option costs $0.40 per ounce, how many ounces of each food should be purchased to minimize costs? What is that minimum cost (per day)? - The number of ounces of each food purchased should be given as whole number values and the minimum cost should be frounded to two decimal places.

Desired Output

`![image-2.png](attachment:image-2.png)`

Solution to Problem 5

In [24]:
```
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize

# Declaring variables
dairybased = LpVariable("DairyBased", lowBound=0)
veganbased = LpVariable("VeganBased", lowBound = 0)

#Defining the problem
problem = LpProblem("problem",LpMinimize)

#Defining the constraints
problem += (50*dairybased) + (20*veganbased) >= 2950 #vitamin C
problem += (40*veganbased) + (30*dairybased) >= 2890 #calcium
problem += (10*dairybased) + (50*veganbased) >= 2430 #calories

#Defining the objective function
problem += (0.33*dairybased) + (0.40 * veganbased)

#solving the problem
solution = problem.solve()
LpStatus[solution]

print(f"You should buy {int(value(dairybased))} ounces dairy based option and {int(val
print(f"The minimum cost is ${(0.33*value(dairybased)) + (0.40 * value(veganbased))}")
```

```
You should buy 43 ounces dairy based option and 40 of the vegan one.
The minimum cost is $30.19
```