Homework 1 (Question 1 : Standard Form)

In [1]:
```
import sys
!{sys.executable} -m pip install pulp
```

```
Requirement already satisfied: pulp in c:\users\muntakim\appdata\local\programs\python\p
ython38-32\lib\site-packages (2.4)
Requirement already satisfied: amply>=0.1.2 in c:\users\muntakim\appdata\local\programs
\python\python38-32\lib\site-packages (from pulp) (0.1.4)
Requirement already satisfied: pyparsing in c:\users\muntakim\appdata\local\programs\pyt
hon\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (2.4.7)
Requirement already satisfied: docutils>=0.3 in c:\users\muntakim\appdata\local\programs
\python\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (0.17.1)
```

In [2]:
```
import pulp
from pulp import *
```

In [3]:
```
"""
Linear Programming Problem 1.0 (Standard Form)

Author : Muntakim Rahman 2021
"""
```

Out[3]:
```
'\nLinear Programming Problem 1.0 (Standard Form)\n\nAuthor : Muntakim Rahman 2021\n'
```

Linear Programming Problem 1.0

$$\begin{aligned} \text{Minimize} \quad & x_1 - 3x_2 - x_3 \\ \text{Subject to} \quad & x_1 + x_2 + x_3 = 3 \\ & -x_1 + x_2 \leq 1 \\ & x_1 \geq 0 \\ & x_2 \; unconstrained \\ & x_3 \geq 0 \end{aligned}$$

Standard Inequality Form

$$\begin{aligned} \text{Maximize} \quad & -x_1 + 3x_2^+ - 3x_2^- + x_3 \\ \text{Subject to} \quad & x_1 + x_2^+ - x_2^- + x_3 \leq 3 \\ & -x_1 - x_2^+ + x_2^- - x_3 \leq -3 \\ & -x_1 + x_2^+ - x_2^- \leq 1 \\ & x_1, x_2^+, x_2^-, x_3 \geq 0 \end{aligned}$$

In [4]:
```
decision_variables__ = {}

decision_variables__['x_1'] = LpVariable(name = 'x_1', lowBound = 0, cat = LpContinuous
decision_variables__['x_2_pos'] = LpVariable(name = 'x_2_pos', lowBound = 0, cat = LpCo
decision_variables__['x_2_neg'] = LpVariable(name = 'x_2_neg', lowBound = 0, cat = LpCo
```

```
decision_variables__['x_3'] = LpVariable(name = 'x_3', lowBound = 0, cat = LpContinuous

## Print Decision Variables -> Mainly for Debugging Purposes.
print(decision_variables__)
```

```
{'x_1': x_1, 'x_2_pos': x_2_pos, 'x_2_neg': x_2_neg, 'x_3': x_3}
```

In [5]:

```
LP_Prob_St = LpProblem(name = 'LP_Problem_1.0_Standard_Form', sense = LpMaximize)

# The Objective Function is Added to 'LP_Prob_St' First.
LP_Prob_St += - decision_variables__['x_1'] + 3 * decision_variables__['x_2_pos'] - 3 *
```

In [6]:

```
# The Constraints are Added to 'LP_Prob_St'
LP_Prob_St += decision_variables__['x_1'] + decision_variables__['x_2_pos'] - decision_
LP_Prob_St += - decision_variables__['x_1'] - decision_variables__['x_2_pos'] + decisio
LP_Prob_St += - decision_variables__['x_1'] + decision_variables__['x_2_pos'] - decisio

LP_Prob_St += decision_variables__['x_1'] >= 0
LP_Prob_St += decision_variables__['x_2_pos'] >= 0
LP_Prob_St += decision_variables__['x_2_neg'] >= 0
LP_Prob_St += decision_variables__['x_3'] >= 0
```

In [7]:

```
print(LP_Prob_St)
```

```
LP_Problem_1.0_Standard_Form:
MAXIMIZE
-1*x_1 + -3*x_2_neg + 3*x_2_pos + 1*x_3 + 0
SUBJECT TO
_C1: x_1 - x_2_neg + x_2_pos + x_3 <= 3

_C2: - x_1 + x_2_neg - x_2_pos - x_3 <= -3

_C3: - x_1 - x_2_neg + x_2_pos <= 1

_C4: x_1 >= 0

_C5: x_2_pos >= 0

_C6: x_2_neg >= 0

_C7: x_3 >= 0

VARIABLES
x_1 Continuous
x_2_neg Continuous
x_2_pos Continuous
x_3 Continuous
```

In [8]:

```
LP_Prob_St.writeLP('LP_ProblemStandardForm.lp')
```

Out[8]:

```
[x_1, x_2_neg, x_2_pos, x_3]
```

In [9]:

```
# The Problem is Solved Using PuLP's Choice of Solver.
LP_Prob_St.solve()
```

Out[9]: 1

In [10]:

```
print(f'Status: {LpStatus[LP_Prob_St.status]} \n')

for variable in LP_Prob_St.variables() :
    print(f'{variable.name} = {variable.varValue}')
print('\n')

if (LpStatus[LP_Prob_St.status] == 'Optimal') :
    print(f'Optimal Value : Z = {value(LP_Prob_St.objective)}')
else :
    print(f'No Optimal Value. Status Code : {value(LP_Prob_St.objective)}')
```

```
Status: Optimal

x_1 = 1.0
x_2_neg = 0.0
x_2_pos = 2.0
x_3 = 0.0


Optimal Value : Z = 5.0
```

In [11]:

```
# Values of original LP Variables are calculated.
x_1 = decision_variables__['x_1'].varValue
x_2 = decision_variables__['x_2_pos'].varValue - decision_variables__['x_2_neg'].varVal
x_3 = decision_variables__['x_3'].varValue
```

In [12]:

```
print(f'x_1 from Original Problem = {x_1}')
print(f'x_2 from Original Problem = {x_2}')
print(f'x_3 from Original Problem = {x_3}')
```

```
x_1 from Original Problem = 1.0
x_2 from Original Problem = 2.0
x_3 from Original Problem = 0.0
```