MP-2 Tutorial - 2

190031920

Avuthu Nikhil Reddy

In [1]:
```
import numpy as np

def func(X):
    x=X[0]
    y=X[1]
    L=X[2]
    return 6*x**2 +12*y**2 -L * (x+y-90)
```

In [2]:
```
def dfunc(X):
    dLambda=np.zeros(len(X))
    h=1e-3
    for i in range(len(X)):
        dX=np.zeros(len(X))
        dX[i]=h
        dLambda[i]=(func(X+dX)-func(X-dX))/(2*h);
    return dLambda
```

In [3]:
```
from scipy.optimize import fsolve
x1=fsolve(dfunc,[1,1,0])
print(x1,func(x1))
x2=fsolve(dfunc,[-1,-1,0])
print(x2,func(x2))
```

```
[ 60.          30.         720.00000001] 32400.0
[ 60.          30.         719.99999999] 32400.0
```

In [4]:
```
def knapSack(W, wt, val, n):
   K = [[0 for x in range(W + 1)] for x in range(n + 1)]
   for i in range(n + 1):
      for w in range(W + 1):
         if i == 0 or w == 0:
            K[i][w] = 0
         elif wt[i-1] <= w:
            K[i][w] = max(val[i-1] + K[i-1][w-wt[i-1]], K[i-1][w])
         else:
            K[i][w] = K[i-1][w]
   return K[n][W]
val = [16,22,12,8]
wt = [5,7,4,3]
W = 14
n = len(val)
print(knapSack(W, wt, val, n))
```

```
42
```