Individual Assignment Report

TASK 1

```
# create decision variables
r1 = m.addVar(name ="first_month_regular")
r2 = m.addVar(name ="second_month_regular")
r3 = m.addVar(name ="third_month_regular")
r4 = m.addVar(name ="fourth_month_regular")
ep1 = m.addVar(name ="first_month_extra")
ep2 = m.addVar(name ="second_month_extra")
ep3 = m.addVar(name ="third_month_extra")
ep4 = m.addVar(name ="fourth_month_extra")

# total production
prod_m1 = r1+ep1
prod_m2 = r2+ep2
prod_m3 = r3+ep3
prod_m4 = r4+ep4
```

To minimizing the total cost

= Total cost of Storage + Total cost of Regular production + Total cost of Extra Production

```
obj = stor_total_cost + rp_total_cost + ep_total_cost
m.setObjective(obj, GRB.MINIMIZE)
```

Calculation of Total cost of storage :

storage of first month is total production of first month - Demand of first month

= regular production of 1 + extra production of 1 - Demand of 1(di)

storage of second month is total production of second month + storage of first month - Demand of second month

= regular production of 2 + extra production of 2+(regular production of 1 + extra production of 1 - Demand of 2(di)) -Demand of 2

storage of thrid month is total production of third month + storage of second month - Demand of third month

=regular production of 3 + extra production of 3+(regular production of 2 + extra production of 2 - Demand of 2(di)) -Demand of 3

storage of fourth month is total production of fourth month + storage of third month - Demand of fourth month.

=regular production of 4 + extra production of 4+(regular production of 3 + extra production of 3 - Demand of 3(di)) -Demand of 4

```
# Storage
stor_m1 = prod_m1-120
stor_m2 = prod_m2+(stor_m1)-160
stor_m3 = prod_m3+(stor_m2)-300
stor_m4 = prod_m4+(stor_m3)-200

# Storage total cost
# sum ( each month storage * $5)
stor_total_cost = (stor_m1 * 5) + (stor_m2 * 5) + (stor_m3 * 5) + (stor_m4 * 5)
```

Calculation of Total cost of Regular production

=total of regular production * Production cost (ci)

= total of regular production * 440

```
# Regular production total cost
# sum( each month regular production * each moth production cost)
# sum( regular procution * ci )
rp_total_cost = (r1+r2+r3+r4) * 440
```

Calculation of Total cost of Extra Production

= total of Extra production * <(Production cost (ci) + Extra production cost (ei)>

```
# Extra production total cost
# sum(Eeach month extra production * each month extra production cost)
ep_total_cost = (ep1+ep2+ep3+ep4) * (440 + 260)
```

TASK 2

Code the problem using Python and Gurobi.

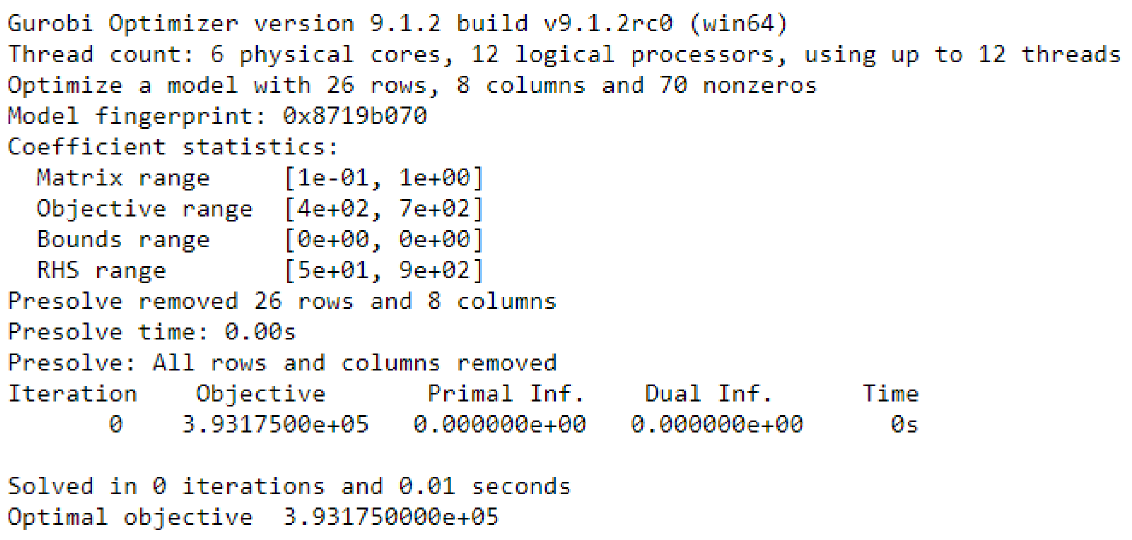
```
import gurobipy as gp
from gurobipy import GRB
import pandas as pd
from collections import OrderedDict
```

Model :

```
# model : Production Planning Problem
m = gp.Model("PPP")

# create decision variables
r1 = m.addVar(name ="first_month_regular")
r2 = m.addVar(name ="second_month_regular")
r3 = m.addVar(name ="third_month_regular")
r4 = m.addVar(name ="fourth_month_regular")
ep1 = m.addVar(name ="first_month_extra")
ep2 = m.addVar(name ="second_month_extra")
```

```
ep3 = m.addVar(name ="third_month_extra")
ep4 = m.addVar(name ="fourth_month_extra")

# total production
prod_m1 = r1+ep1
prod_m2 = r2+ep2
prod_m3 = r3+ep3
prod_m4 = r4+ep4

# Storage
stor_m1 = prod_m1-120
stor_m2 = prod_m2+(stor_m1)-160
stor_m3 = prod_m3+(stor_m2)-300
stor_m4 = prod_m4+(stor_m3)-200

# Storage total cost
# sum ( each month storage * $5)
stor_total_cost = (stor_m1 * 5) + (stor_m2 * 5) + (stor_m3 * 5) + (stor_m4 * 5)

# Regular production total cost
# sum( each month regular production * each moth production cost)
# sum( regular procuction * ci )
rp_total_cost = (r1+r2+r3+r4) * 440

# Extra production total cost
# sum(Eeach month extra production * each month extra production cost)
ep_total_cost = (ep1+ep2+ep3+ep4) * (440 + 260)

# the objective : minimize the cost
    #storage --> previous month...
# sum(each month storage * 5 dollar) + sum(regular proudction total cost) + sum(extra production cost)

obj = stor_total_cost + rp_total_cost + ep_total_cost
m.setObjective(obj, GRB.MINIMIZE)

########################
# constraints below;:::#
########################

# maxium storage availability
con10 = m.addConstr(stor_m1 <=100, name ='storage_1st_month')
con11 = m.addConstr(stor_m2 <=100,name ="storage_2nd_month")
con12 = m.addConstr(stor_m3 <=100,name ="storage_3rd_month")
con13 = m.addConstr(stor_m4 <=100,name ="storage_4th_month")
# con14 = m.addConstr(stor_m1+stor_m2+stor_m3+stor_m4 <= 100, name ='total_storage_capacity_100')
# end of 4th month storage must be zero.
con15 = m.addConstr(stor_m4 == 0, name ="month_4th_zero")


# regular production in each month must be last 10% of total production of first three month okay
con20 = m.addConstr(r1+r2+r3 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)

# CONFUSED WITH IT BETWEEN ABOVE VS BELOW
#    con21 = m.addConstr(r1 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con22 = m.addConstr(r2 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con23 = m.addConstr(r3 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con24 = m.addConstr(r4 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)


# regular production less than max production pi okay
con31 = m.addConstr(r1 <= 140)
con32 = m.addConstr(r2 <= 150)
con33 = m.addConstr(r3 <= 140)
con34 = m.addConstr(r4 <= 160)

# Extra production less than max extra production qi okay
```

```
con40 = m.addConstr(ep1 <= 50)
con41 = m.addConstr(ep2 <= 75)
con42 = m.addConstr(ep3 <= 70)
con43 = m.addConstr(ep4 <= 80)

# non-negativeity constraints okay
con70 = m.addConstr(r1>=0)
con71 = m.addConstr(r2>=0)
con72 = m.addConstr(r3>=0)
con73 = m.addConstr(r4>=0)
con74 = m.addConstr(ep1>=0)
con75 = m.addConstr(ep2>=0)
con76 = m.addConstr(ep3>=0)
con77 = m.addConstr(ep4>=0)
con78 = m.addConstr(stor_m1 >= 0)
con781 = m.addConstr(stor_m2 >= 0)
con782 = m.addConstr(stor_m3 >= 0)
con783 = m.addConstr(stor_m4 >= 0)

#solve
m.optimize()
```



```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads
Optimize a model with 26 rows, 8 columns and 70 nonzeros
Model fingerprint: 0x8719b070
Coefficient statistics:
  Matrix range     [1e-01, 1e+00]
  Objective range  [4e+02, 7e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [5e+01, 9e+02]
Presolve removed 26 rows and 8 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.9317500e+05   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds
Optimal objective  3.931750000e+05
```

Prink Optimal Value of Object function and decision variables

```
# Print optimal value of the objective function
print('\nTotal Operating Cost: %g' %m.objVal)
# Print optimal values for the decision variables
print('\nDecision variables:')
for v in m.getVars():
    print('%s = %g' % (v.varName, v.x))
```

TASK 3

Solve the problem that you have coded and report the optimal monthly production amounts and total cost

Total Cost was $393,175

Optimal Monthly Production Amounts :

first month

Regular Production : 140 tons

Extra Production: 5 tons

second month

Regular Production : 150 tons

Extra Production: 75 tons

third month

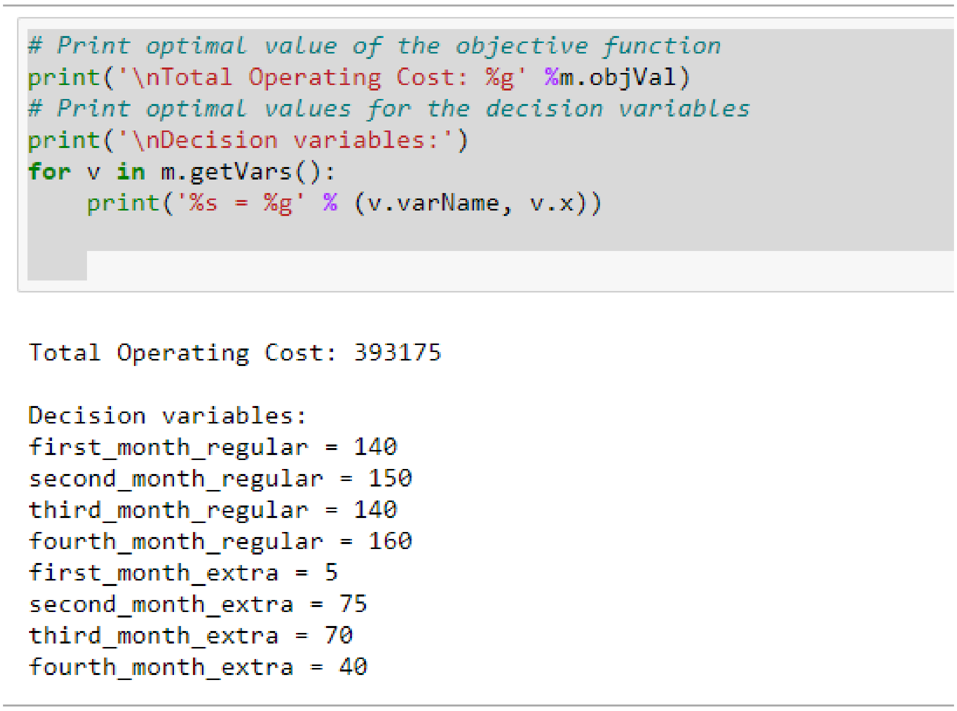
Regular Production : 140 tons

Extra Production: 70 tons

fourth month

Regular Production : 160 tons

Extra Production: 40 tons



```
# Print optimal value of the objective function
print('\nTotal Operating Cost: %g' %m.objVal)
# Print optimal values for the decision variables
print('\nDecision variables:')
for v in m.getVars():
    print('%s = %g' % (v.varName, v.x))
```

```
Total Operating Cost: 393175

Decision variables:
first_month_regular = 140
second_month_regular = 150
third_month_regular = 140
fourth_month_regular = 160
first_month_extra = 5
second_month_extra = 75
third_month_extra = 70
fourth_month_extra = 40
```