In [2]:

```
import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

In [3]:

```
modelo = Model(name='Lista_1_Questao_3')
```

In [5]:

```
trigo = modelo.continuous_var(name='Produção em kg de trigo')
arroz = modelo.continuous_var(name='Produção em kg de arroz')
milho = modelo.continuous_var(name='Produção em kg de milho')
```

In [6]:

```
# Função Objetiva
modelo.maximize(10.8*trigo + 4.2*arroz + 2.03*milho)
```

In [7]:

```
# Restrições
modelo.add_constraint(trigo >= 0)
modelo.add_constraint(arroz >= 0)
modelo.add_constraint(milho >= 0)
modelo.add_constraint(trigo/0.2 + arroz/0.3 + milho/0.4 <= 200000)
modelo.add_constraint(trigo + arroz + milho <= 60000)
modelo.add_constraint(trigo/0.2 >= 400)
modelo.add_constraint(arroz/0.3 >= 800)
modelo.add_constraint(milho/0.4 >= 10000)
```

Out[7]:

```
docplex.mp.LinearConstraint[](2.500Produção em kg de milho,GE,10000)
```

In [8]:

```
modelo.print_information()
```

```
Model: Lista_1_Questao_3
 - number of variables: 3
   - binary=0, integer=0, continuous=3
 - number of constraints: 8
   - linear=8
 - parameters: defaults
 - objective: maximize
 - problem type is: LP
```

In [9]:

```
otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 417800.000
  "Produção em kg de trigo"=37840.000
  "Produção em kg de arroz"=240.000
  "Produção em kg de milho"=4000.000
```

In [10]:

```
modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck                          1
CPXPARAM_LPMethod                                4
Tried aggregator 1 time.
LP Presolve eliminated 6 rows and 0 columns.
Reduced LP has 2 rows, 3 columns, and 6 nonzeros.
```

```
Presolve time = 0.00 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 1
Using Approximate Minimum Degree ordering
Total time for automatic ordering = 0.02 sec. (0.00 ticks)
Summary statistics for Cholesky factor:
  Threads                   = 4
  Rows in Factor            = 2
  Integer space required    = 2
  Total non-zeros in factor = 3
  Total FP ops to factor    = 5
 Itn      Primal Obj        Dual Obj  Prim Inf Upper Inf  Dual Inf Inf Ratio
   0   3.2009655e+05  -5.0976000e+04  0.00e+00  0.00e+00  7.60e+01  1.00e+00
   1   3.4465240e+05   2.8762128e+05  0.00e+00  0.00e+00  1.07e+01  6.90e+00
   2   4.1628196e+05   4.1921548e+05  4.19e-09  0.00e+00  1.78e-01  4.30e+02
   3   4.1779954e+05   4.1779991e+05  3.10e-11  0.00e+00  2.82e-05  2.62e+06
   4   4.1780000e+05   4.1780000e+05  6.98e-11  0.00e+00  2.82e-09  2.40e+10
   5   4.1780000e+05   4.1780000e+05  4.65e-11  0.00e+00  2.81e-13  2.40e+14
Barrier time = 0.03 sec. (0.01 ticks)
Parallel mode: deterministic, using up to 4 threads for concurrent optimization:
 * Starting dual Simplex on 1 thread...
 * Starting primal Simplex on 1 thread...

Dual crossover.
  Dual:  Fixed no variables.
  Primal:  Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.03 sec. (0.00 ticks)

Total time on 4 threads = 0.06 sec. (0.01 ticks)
```

```
Out[10]: docplex.mp.solution.SolveSolution(obj=417800,values={Produção em kg de t..
```

```
In [11]: %notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_3.ipynb"
```