1)

X1 = Atlanta - Los Angeles

X2 = Atlanta - New York

X3 = Tulsa - Los Angeles

X4 = Tulsa - New York

X5 = Seattle - Los Angeles

X6 = Seattle - New York

X7 = Baltimore- Los Angeles

X8 = Baltimore - New York

Objective Function: Minimize $8X1 + $5X2 + $4X3 + $7X4 + $5X5 + $6X6 + $4X7 + $6X8

$X1 + X2 <= 600$

$X3 + X4 <=900$

$X5 + X6 <= 500$

$X7+ X8 <=500$

$X1 + X3 +X5 +X7 <=800$

$X2 + X4 +X6 +X8 <=1200$

$X1, X2, X3, X4, X5, X6, X7, X8 >=0$

	A	B	C	D	E	F
1						
2	**Retail Stores**	Manufacturing Plant			**Demand**	
3		Atlanta	Tulsa	Seattle		
4	Los Angeles	$8.00	$4.00	$5.00	800	
5	New York	$5.00	$7.00	$6.00	1200	
6	**Capacity**	600	900	500		
7						
8	**Retail Stores**	Manufacturing Plant			**Demand**	
9		Atlanta	Tulsa	Seattle		
10	Los Angeles				0	
11	New York				0	
12	**Capacity**	0	0	0		
13						
14						
15	**Retail Stores**	Manufacturing Plant			**Demand**	
16		Atlanta	Tulsa	Seattle		
17	Los Angeles	$0.00	$0.00	$0.00		
18	New York	$0.00	$0.00	$0.00		
19	**Capacity**				$0.00	Total cost

Retail Stores	Manufacturing Plant			Demand
	Atlanta	Tulsa	Seattle	
Los Angeles	$8.00	$4.00	$5.00	800
New York	$5.00	$7.00	$6.00	1200
Capacity	600	900	500	

Retail Stores	Manufacturing Plant			Demand
	Atlanta	Tulsa	Seattle	
Los Angeles	0	800	0	800
New York	600	100	500	1200
Capacity	600	900	500	

Retail Stores	Manufacturing Plant			Demand	
	Atlanta	Tulsa	Seattle		
Los Angeles	$0.00	$3,200.00	$0.00		
New York	$3,000.00	$700.00	$3,000.00		
Capacity				$9,900.00	Total cost

The total cost in choosing Seattle is $9900.

	A	B	C	D	E	F
1						
2	**Retail Stores**	Manufacturing Plant			**Demand**	
3		Atlanta	Tulsa	Baltimore		
4	Los Angeles	$8.00	$4.00	$4.00	800	
5	New York	$5.00	$7.00	$6.00	1200	
6	**Capacity**	600	900	500		
7						
8	**Retail Stores**	Manufacturing Plant			**Demand**	
9		Atlanta	Tulsa	Baltimore		
10	Los Angeles	0	800	0	800	
11	New York	600	100	500	1200	
12	**Capacity**	600	900	500		
13						
14	**Retail Stores**	Manufacturing Plant			**Demand**	
15		Atlanta	Tulsa	Baltimore		
16	Los Angeles	$0.00	$3,200.00	$0.00		
17	New York	$3,000.00	$700.00	$3,000.00		
18	**Capacity**				$9,900.00	Total cost

The total cost in choosing Baltimore is $9900.

The cost is same in both the cases. Therefore, **the company can select either of the location.**

2)

Variables:

X1=Beef

X2=Pork

X3=Chicken

X4-Turkey

Objective function:

Objective is to provide most economical combination of four meats to make this hot dog i.e.

MinZ = $0.76X1+0.82X2+0.64X3+0.58X4$

Constraints:

$X1+X2+X3+X4=0.125$	**Constraints for Weight of each hot dog (2 ounce= 0.125 pound)**
$32.5X1+54X2+25.6X3+6.4X4 <= 6$	**Constraints for grams of Fat**
$210X1+205X2+220X3+172X4 <= 27$	**Constraints for grams of Cholesterol**
$640X1+1055X2+780X3+528X4 <= 100$	**Constraints for grams of Calorie**
$X1>=0.25(X1+X2+X3+X4)$	**Constraint for each 2-ounce hot dog to be at least 25% beef**
$X2>=0.25(X1+X2+X3+X4)$	**Constraint for each 2-ounce hot dog to be at least 25% pork**
$X1,X2,X3, X4 >=0$	**Non-negativity constraints**

	Cost/Pound	Calories/Pound	Fat (G/lb)	Cholesterol (G/lb)	Lb of meat/Hotdog
Beef(X1)	$ 0.76	640	32.5	210	0.03125
Pork(X2)	$ 0.82	1055	54	205	0.03125
Chicken(X3)	$ 0.64	780	25.6	220	0
Turkey(X4)	$ 0.58	528	6.4	172	0.0625
Objective Function (MinZ)	0.085625				
Constraints					
Weight Constraint	0.125	=	0.125		
Fat Constraint	3.103125	<=	6		
Cholesterol Constraint	23.71875	<=	27		
Calorie Constraint	85.96875	<=	100		
Constraints 25% beef	0.03125	>=	0.03125		
Constraints 25% pork	0.03125	>=	0.03125		

3)

sij	1	2	3	4	5
1	0	20	15	8	6
2	15	0	18	9	28
3	24	23	0	13	13
4	15	27	8	0	14
5	8	17	24	15	0

Total Surgical Setup Time 58

xij	1	2	3	4	5	
1	0	1	0	0	0	1
2	0	0	0	1	0	1
3	0	0	0	0	1	1
4	0	0	1	0	0	1
5	1	0	0	0	0	1
	1	1	1	1	1	

1-2-4-3-5-1 **Optimal Schedule (no Sub-Schedules)**

Sub-Schedule Elimination Constraints

Sub-Schedule: 1-5

x1,5	x5,1			
0	1	1	<=	1

sij	1	2	3	4	5	6	7	8	9	10
1	0	9	12	26	11	24	12	13	17	15
2	24	0	28	23	22	5	7	18	9	23
3	19	30	0	30	15	22	25	15	28	15
4	18	10	27	0	28	12	16	19	22	7
5	5	16	11	7	0	25	27	30	23	15
6	7	26	6	17	6	0	28	10	13	28
7	23	26	20	20	24	30	0	16	18	27
8	23	20	22	8	18	10	14	0	14	12
9	7	13	9	19	29	27	18	23	0	30
10	16	10	11	11	28	26	6	11	12	0

Total Surgical Setup Time 92

xij	1	2	3	4	5	6	7	8	9	10	
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	1	0	0	0	0	1
3	0	0	0	0	1	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	1	1
5	1	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	1	0	0	1
7	0	0	0	0	0	0	0	0	1	0	1
8	0	0	0	1	0	0	0	0	0	0	1
9	0	0	1	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	1	0	0	0	1
	1	1	1	1	1	1	1	1	1	1	

1-2-6-8-4-10-7-9-3-5-1 Optimal Schedule (no Sub-Schedules)

Sub-Schedule Elimination Constraints

Sub-Schedule: 1-2-6-5

x1,2	x2,1	x2,6	x6,2	x6,5	x5,6	x5,1	x1,5			
1	0	1	0	0	0	1	0	3	<=	3

Sub-Schedule: 1-2-9

x1,2	x2,1	x2,9	x9,2	x9,1	x1,9			
1	0	0	0	0	0	1	<=	2

4)

Goals	Description	Rank	Wt
Goal 1	Minimize overutilization of plastic	2	0.364
Goal 2	Minimize overutilization of metals	1	0.182

Goal 3	Minimize overutilization of rubber	1	0.182
Goal 4	Minimize overutilization of budget	0.5	0.091
Goal 5	Minimize underutilization of budget	0.5	0.091
Goal 6	Maximize available hours usage (i.e. Minimize underutilization)	0.5	0.091

Decision Variables:

X1 = No. of Tiny Tanks produced

X2 = No. of Tiny Trucks produced

X3 = No. pf Tiny Turle produced

'Oi' = Excess of right side of the goal w.r.t. the goal 'i' and

'Ui' = Deficit of right side of the goal w.r.t. the goal 'i' for all i = 1,2,...,6

Objective Function:

Min (0.364O1 + 0.182O2 + 0.182O3 + 0.091O4 + 0.091U5 + 0.091U6)

Subject to,

1.5X1 + 2.0X2 + 1.0X3 + U1 - O1 = 16000

0.5X1 + 0.5X2 + 1.0X3 + U3 - O3 = 5000

0.3X1 + 0.6X2 + 0.0X3 + U2 - O2 = 9000

2.0X1 + 2.0X2 + 1.0X3 + U6 - O6 = 40

7.0X1 + 5.0X2 + 4.0X3 + U4 - O4 = 164000

7.0X1 + 5.0X2 + 4.0X3 + U5 - O5 = 164000

X1, X2, X3, Oi, Ui >= 0

5)

Let X_1 = thousands of dollars spent on TV ads

X_2 = thousands of dollars spent on radio ads

Min $P_1(d_1^-) + P_2(d_2^+) + P_3(d_3^+) + P_4(d_4^-) + P_5(d_5^- + 2d_6^-)$

Subject to:

$10{,}000X_1 + 7{,}500X_2 + d_1^- - d_1^+ = 750{,}000$	(At least 750,000 exposures)
$X_1 + X_2 + d_2^- - d_2^+ = 100$	(At most \$100,000 spent)
$X_1 + d_3^- - d_3^+ = 70$	(At most \$70,000 spent on TV ads)
$10{,}000X_1 + 7{,}500X_2 + d_4^- - d_4^+ = 1{,}000{,}000$	(Achieve 1 million exposures)
$2{,}500X_1 + 3{,}000X_2 + d_5^- - d_5^+ = 250{,}000$	(At least 250,000 18-21 exposures)
$3{,}000X_1 + 1{,}500X_2 + d_6^- - d_6^+ = 250{,}000$	(At least 250,000 25-30 exposures)

$d_i^+ \geq 0 \quad and \quad d_i^- \geq 0 \quad \forall i \quad i = 1,2,3,4,5,6$

$X_j \geq 0 \quad \forall j \quad j = 1,2$

Extra Credit:

a)

Pattern	3-ft	4-ft	5-ft
1	3	0	0
2	2	1	0
3	1	0	1
4	0	1	1
5	0	2	0
6	0	0	2

b)

Define x_i as the number of 10-ft boards cut into pattern type i, $i = 1, \ldots, 6$. Then:

$$
\begin{array}{rl}
\min & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
\text{subject to} & 3x_1 + 2x_2 + 1x_3 \geq 90 \\
 & 1x_2 + 1x_4 + 2x_5 \geq 60 \\
 & 1x_3 + 1x_4 + 2x_6 \geq 60 \\
 & x_1, x_2, x_3, x_4, x_5, x_6 \in \mathbb{Z}_+
\end{array}
$$

The minimal number of 10-ft boards to be cut is 83. This can be done by cutting 2 boards into pattern 1, 42 into pattern 2, 9 into pattern 5, and 30 into pattern 6 (among other ways -- there are multiple optimal solutions to this integer program). Incidentally, this solution produces 20 feet of scrap.

Original Length	10

	Board width (in feet)					
Pattern #	**3**	**4**	**5**	**Waste (ft)**	**Pattern**	**To Cut**
1	3	0	0	1	1	2
2	2	1	0	0	2	42
3	1	0	1	2	3	0
4	0	1	1	1	4	0
5	0	2	0	2	5	9
6	0	0	2	0	6	30

	3	4	5
	90	60	60
	>=	>=	>=
Demand	90	60	60
Leftover Inventory	0	0	0

Min Total Cut	83
Excess Scrap	20
Excess Inventory	0
Total Excess	20

c)

```
Optimize a model with 3 rows, 6 columns and 9 nonzeros
Coefficient statistics:
  Matrix range    [1e+00, 3e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [0e+00, 0e+00]
  RHS range       [6e+01, 9e+01]
Found heuristic solution: objective 120
Presolve time: 0.00s
Presolved: 3 rows, 6 columns, 9 nonzeros
Variable types: 0 continuous, 6 integer (0 binary)

Root relaxation: objective 8.250000e+01, 3 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0   82.50000    0    1  120.00000   82.50000  31.3%     -    0s
H    0     0                       83.0000000   82.50000  0.60%     -    0s

Explored 0 nodes (3 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Optimal solution found (tolerance 1.00e-04)
Best objective 8.300000000000e+01, best bound 8.300000000000e+01, gap 0.0%

Minimum # of Rolls = 83.0
Cut 45.0 of pattern2
Cut 8.0 of pattern5
Cut 30.0 of pattern6
```

```python
nWidths = 3;
nPatterns = 6;

widthsInPatterns = [
[3, 2, 1, 0, 0, 0],
[0, 1, 0, 1, 2, 0],
[0, 0, 1, 1, 0, 2]
];
demand = [90, 60, 60];

model = Model("lumberyard")

patternsCut = []

for i in range(nPatterns):
    patternsCut.append(model.addVar(vtype=GRB.INTEGER, obj=1, lb=0,
    name="pattern%d" % (i+1)))

model.modelSense = GRB.MINIMIZE
model.update()

expr = LinExpr()
for i in range(nWidths):
    expr = 0
    for j in range(nPatterns):
        expr += widthsInPatterns[i][j]*patternsCut[j]
    model.addConstr(expr, GRB.GREATER_EQUAL,demand[i] , name="sum%d" %(i))
model.update()
model.write('lumberyard.lp')
model.optimize()

if model.Status == GRB.OPTIMAL:
    print "\nMinimum # of Rolls =", model.ObjVal
    for v in model.getVars():
        if v.X > 0:
            print "Cut", v.X, "of", v.VarName

```