Problem 1 (20 %)

A manufacturing company produces three products A, B, and C, which are processed on three machines.

For machine 1, there are 400 hours available. For machine 2, there are 300 hours available. For machine 3, there are 500 hours available.

For each product and each machine, the following table shows how many hours are required to produce one unit and the profit per unit of the products:

	Machine 1	Machine 2	Machine 3	Profit
A	4	3	4	65
B	7	3	5	80
C	2	6	5	70

a) Use the given data and formulate a model to maximize total profit, subject to the capacity constraints. (You can choose either a LINGO model or a mathematical model.)

b) Extend the model to take into account the following additional information: Because of market constraints, the company needs to choose two out of the three products.

a)

Mathematical model:

Max $65\ X_A + 80\ X_B + 70\ X_C$

S.t. $4\ X_A + 7\ X_B + 2\ X_C \le 400$

$3\ X_A + 3\ X_B + 6\ X_C \le 300$

$4\ X_A + 5\ X_B + 5\ X_C \le 500$

$X_A\ ,\ X_B\ ,\ X_C \ge 0$

X_i = production quantity for product *i*

b)

Mathematical model:

Max $65\ X_A + 80\ X_B + 70\ X_C$

S.t. $4\ X_A + 7\ X_B + 2\ X_C \le 400$

$3\ X_A + 3\ X_B + 6\ X_C \le 300$

$4\ X_A + 5\ X_B + 5\ X_C \le 500$

$X_A \le M_A Y_A$

$X_B \le M_B Y_B$

$X_C \le M_C Y_C$

$Y_C + Y_C + Y_C \le 2$

$X_A\ ,\ X_B\ ,\ X_C \ge 0$

$Y_A\ ,\ Y_B\ ,\ Y_C = 0/1$

Y_i = 1 if product *i* is produced, and 0 otherwise.

M_i = "Big M", a number greater than or equal to maximum production quantity for product i.

Problem 2 (35 %)

Below is an AMPL model for facility location.

```
param N;
param P;
param population {1..N};
param distance {1..N, 1..N};
var Locate {1..N} binary;
var Assign {1..N, 1..N} binary;

minimize TotalDistance:
sum {i in 1..N, j in 1..N}
      population[i] * distance[i,j] * Assign [i,j];

s.t. constr1:
      sum {j in 1..N} Locate[j] = P;

s.t. constr2 {i in 1..N}:
      sum {j in 1..N} Assign[i,j] = 1;

s.t. constr3 {i in 1..N, j in 1..N}:
      Assign[i,j] - Locate[j] <= 0;
```

Consultants are trying to use this model to support the decisions on where to locate fire stations in a region.

a) In such a setting, give a possible explanation for each constraint and the objective function of the above model.
Objective function:
minimize total weighted distance between fire stations and the population.
Constr1:
Exactly P facilities should be located.
Constr2:
Every population node must be assigned to one fire station.
Constr3:
A population node cannot be assigned to a potential fire station location unless a fire station is located there.

b) Suggest how the model can be modified to take into account an upper limit on the number of nodes that can be covered by each fire station.

```
param maxnumber;
s.t. constr4 {j in 1..N}:
      sum {i in 1..N} Assign[i,j] <= maxnumber;
```

c) Suggest how the model can be modified to take into account an upper limit on the area (in terms of population) that can be covered by each fire station.

```
param maxpopulation;
s.t. constr4 {j in 1..N}:
      sum {i in 1..N} population[i] * Assign[i,j] <= maxpopulation;
```

d) Suggest how the model can be modified to minimize the maximum distance from a fire station to the population.

```
# New variable:
var MaxDistance;

# Changed objective function:
minimize MaxDistance:

# New constraints:
s.t. constr5 {i in 1..N, j in 1..N}:
      MaxDistance >= distance[i,j] * Assign [i,j];
```

Problem 3 (35 %)

The following shows a formulation for a production planning problem known as the Capacitated Lot Sizing Problem for one product:

$$\text{Min } z = \sum_t h_t \cdot I_t + \sum_t v_t \cdot X_t + \sum_t s_t \cdot Y_t \qquad (1)$$

Subject to

$$I_t = I_{t-1} + X_t - d_t \quad \text{for all } t \qquad (2)$$

$$X_t \le K_t \cdot Y_t \quad \text{for all } t \qquad (3)$$

$$Y_t \in \{0,1\} \quad \text{for all } t \qquad (4)$$

$$X_t \ge 0 \quad \text{for all } t \qquad (5)$$

$$I_t \ge 0 \quad \text{for all } t \qquad (6)$$

h_t is inventory holding cost per unit on inventory per period.

v_t is variable production cost per unit produced.

s_t is setup cost per setup.

d_t is demand per period.

K_t is production capacity per period.

a) Build an AMPL model that represents the above problem.

```
param T;
param h {t in 1..T};
param v {t in 1..T};
param s {t in 1..T};
param k {t in 1..T};
param d {t in 1..T};
param init_inv;
var I {t in 1..T} >=0;
var X {t in 1..T} >=0;
var Y {t in 1..T} binary;
```

```
minimize z:
    sum {t in 1..T} h[t] * I[t] + sum {t in 1..T} v[t] * X[t]
  + sum {t in 1..T} s[t] * Y[t];

subject to Con2a :
    I[1] = init_inv + X[1] - d[1];

subject to Con2b {t in 2..T}:
    I[t] = I[t-1] + X[t] - d[t];

subject to Con3 {t in 1..T}:
    X[t] <= k[t] * Y[t];
```

b) Extend the AMPL model to include multiple products, which use the same production capacity.

```
param T;
param P;
param h {t in 1..T, p in 1..P};
param v {t in 1..T, p in 1..P};
param s {t in 1..T, p in 1..P};
param k {t in 1..T};
param d {t in 1..T, p in 1..P};
param init_inv {p in 1..P};
param a {p in 1..P};

var I {t in 1..T, p in 1..P} >=0;
var X {t in 1..T, p in 1..P} >=0;
var Y {t in 1..T, p in 1..P} binary;

minimize z:
    sum {t in 1..T, p in 1..P} h[t,p] * I[t,p]
  + sum {t in 1..T, p in 1..P} v[t,p] * X[t,p]
  + sum {t in 1..T, p in 1..P} s[t,p] * Y[t,p];

subject to Con2a {p in 1..P}:
    I[1,p] = init_inv[p] + X[1,p] - d[1,p];

subject to Con2b {t in 2..T, p in 1..P}:
    I[t,p] = I[t-1,p] + X[t,p] - d[t,p];

subject to Con3 {t in 1..T, p in 1..P}:
    X[t,p] <= k[t] * Y[t,p];

subject to Con4 {t in 1..T}:
    sum {p in 1..P} a[p] * X[t,p] <= k[t];
```

P = number of products

a[p] = capacity consumption per unit produced of product p

c) For the multi-product model, extend the AMPL model to include a setup time that consumes production capacity each time a product is produced.

```
# New parameter b = setup time for product p
param b {p in 1..P};

# Modified constraint 4:
subject to Con4 {t in 1..T}:
        sum {p in 1..P} a[p] * X[t,p]
      + sum {p in 1..P} b[p] * Y[t,p]  <= k[t];
```

d) The model in c) is a starting point for a company that wants to optimise its campaign planning. A campaign means that the company reduces the price of a product in a given period. The price reduction will then lead to increased demand for the given product in the given period. Imagine that a campaign implies a price reduction of 15%, and that the corresponding demand then increases by 40%. For simplicity, assume that demand for other periods and other products is not affected. Assume that the company has decided to run a total of 5 campaigns during the planning horizon (but that it has not been decided for which products and which periods the campaigns will be run). Extend the above model so that it will simultaneously suggest a production plan and the products and periods for which the campaigns should be run, so that total profit is maximized.

```
# New parameter price[p] = ordinary price for product p
# New variable W[t,p] = run campaign in period t for product p
# New variable price_new[t,p] = price in period t for product p after discount for campaign (price_new = price in most cases)
# New variable d_new[t,p] = demand in period t for product p after increase due to campaign (d_new = d in most cases)

param T;
param P;
param h {t in 1..T, p in 1..P};
param v {t in 1..T, p in 1..P};
param s {t in 1..T, p in 1..P};
param k {t in 1..T};
param d {t in 1..T, p in 1..P};
param init_inv {p in 1..P};
param a {p in 1..P};
param b {p in 1..P};
param price {p in 1..P};
```

```
var I {t in 1..T, p in 1..P} >=0;
var X {t in 1..T, p in 1..P} >=0;
var Y {t in 1..T, p in 1..P} binary;
var W {t in 1..T, p in 1..P} binary;
var price_new {t in 1..T, p in 1..P};
var d_new {t in 1..T, p in 1..P};

maximize z:

    sum {t in 1..T, p in 1..P} price_new[t,p] * d_new[t,p]
  - sum {t in 1..T, p in 1..P} h[t,p] * I[t,p]
  - sum {t in 1..T, p in 1..P} v[t,p] * X[t,p]
  - sum {t in 1..T, p in 1..P} s[t,p] * Y[t,p];

subject to Con2a {p in 1..P}:
    I[1,p] = init_inv[p] + X[1,p] - d_new[1,p];

subject to Con2b {t in 2..T, p in 1..P}:
    I[t,p] = I[t-1,p] + X[t,p] - d_new[t,p];

subject to Con3 {t in 1..T, p in 1..P}:
    X[t,p] <= k[t] * Y[t,p];

subject to Con4 {t in 1..T}:
    sum {p in 1..P} a[p] * X[t,p]
  + sum {p in 1..P} b[p] * Y[t,p]
     <= k[t];

subject to Con5 {t in 1..T, p in 1..P}:
    d_new[t,p] = d[t,p] + 0.4 * d[t,p] * W[t,p];

subject to Con6 {t in 1..T, p in 1..P}:
    price_new[t,p] = price[p] - 0.15 * price[p] * W[t,p];

subject to Con7 :
    sum {t in 1..T, p in 1..P} W[t,p] = 5;
```