PROBLEM 1.

Each watchman in a certain art gallery is on duty during some continuous time period. The Guard Schedule is defined to be a set of pairs [T1(i),T2(i)] - the starting and the ending times of the i'th watchman's duty. Given a Guard Schedule you are required:

(a) To check whether there are at least two watchman in the gallery at each moment of the period [0, EndTime]. If the condition (a) is not fulfilled,

(b) Determine all the periods when the guard is insufficient (less then two watchmen on duty).

(c) Find the minimal number of additional watchmen with duties of a prescribed equal length needed to obtain a valid Guard Schedule, i.e. one with condition (a) fulfilled.

INPUT:

(All times are given in integer minutes.)

EndTime - the time when the guard is over, i.e. the gallery should be guarded within the period [0, EndTime].

N - the number of watchmen.

T1[i]. T2[i], i=1, ..., N - the starting and the ending times of the i'th watchman's duty.

Length - the prescribed length of the duty for each additional watchman.

OUTPUT:

(1) The answer for point (a) in the form "Yes/No".

(2) If the previous answer is "no", the list of pairs (k,1) - the beginnings and the ends of all time periods with insufficient guard, and the number of watchmen in each (0 or 1).

(3) The number of additional watchmen and the list of starting and ending times of every additional watchman's duty.

PROBLEM 2.

N segments are given on the plane by the co-ordinates of their endpoints, N>0. Endpoints of segment are specified by two pairs (x1[i], y1[i]), and (x2[i], y2[i]), 1<=i<=N. The endpoints of any segment are belong to it.

You are required:

1. To organize inputting the data in the form kind of

```
<Enter N -- the number of segments :>
<Enter co-ordinates of i'th segment:>
x1[1]--> y1[1]-->
x2[1]--> y2[1]-->
  :::      :::
```

2. To find a straight line which has common points with as many segments as possible. Any of the common points is allowed to be an endpoint of a segment. To output in increasing order the number of the segments that have common points with found straight line.

***PROBLEM 3.
Nodes numbered by 1, 2, ..., N (N<=50) are connected by a network of roads, each of which is of length 1. The roads are going at different heights and are intersected at the nodes only. At the initial moment 0 there are robots in some of the nodes. The total number of the robots is M (M = 2 or 3). The robots keep moving continuously from one node to another independently and can change the direction of their moving at the nodes only. The robots are not allowed to stop. The speed of the i'th robot equals speed[i] (speed[i] = 1 or 2). The robots are being controlled in such way as to minimize the time all of the robots need to get together at the same place.
You are required to find the minimal time T after which the meeting of all the robots at the same place can occur and to indicate this time T, or else to determine that the meeting of all M robots at the same place is impossible at any time t>=0.
The form of the input should be:
<Input N:>
<Input number of roads K:>
<Road 1 connects points:>
    ...
<Road K connects points:>
    (pairs are input as I J)
<Input number of robots M:>
<Input speed of robot 1:>
    ...
<Input speed of robot M:>
<Input initial position of robot 1:>
    ...
<Input initial position of robot M:>
    (All numbers above must be non-negative integers.)
The form of output is
<Time = ...>

PROBLEM 4.
All streets in a certain rectangular-shaped city situated in an uneven area are going either from south to north (N streets) or from west to east (M streets), so that the city is divided into square blocks with sides equal to 1. Every segment of a street enclosed between two neighbouring crossings goes either only down or only up, or else it may be horizontal.

Matrix K[y,x] (with dimension MxN) contains the heights of the crossings above the sea level.

```
     /\ Y
     |
M    +---+---+---+---+---+---+
     |   |   |   |   |   |   |
     +---+---+---+---+---+---+
.    |   | A |   |   |   |   |
.    +---+---o---+---+---+---+
.    |   |   |   |   |   |   |
     +---+---+---+---+---+---+
     |   |   |   |   | B |   |
2    +---+---+---+---*---+---+
     |   |   |   |   |   |   |
1    +---+---+---+---+---+---+--------->X
     1   2      ...          N
```

You are required to write a program that:

1. Inputs dimension of matrix - M and N.
2. Inputs the matrix elements H[i,j]. i=1,M , j=1,N.
3. Inputs the coordinates of two crossings - A and B.
4. Reports the answer to the question whether it is possible to move from A to B or from B to A, going down all the time.

   If the answer to the question from P.3 proves to be affirmative, the program also
5. Determines at least one such path and displays the coordinates of the crossings involved on the screen.
6. Determines all paths of this kind.