4th Balkan Olympiad in Informatics
Nicosia, Cyprus, 19-25 October 1997

Day 1 - Problem 1 (Compatible fishes)

If you want to create an aquarium with exotic fish you must take the advise of an expert in order to avoid a situation in which a certain type of fish can not coexist and live together with another fish type. The reason is the "incompatibility" among fish types.

Given the number of fish types and the available amount of money for spending, find the maximum number of fish types. For this number of fish types find the maximum amount of money that can be spent, provided that only one fish per type is used.

- **Input:**
  INPUT DATA (file INPUT.TXT) The program reads input lines from the file INPUT.TXT as follows:
  - on the first line two integers, separated by a space are given. The first number represents the maximum available amount of money M (M<=1000) for investment and the second one represents the different number of fish types F (F<=30) available.
  - on F subsequent lines, the identification number of each fish type and the corresponding cost separated by a space, are given.
  - on subsequent lines until the end of file, the file contains pairs of fish types, separated by space, that represent the incompatibilities among the different fish types (those fish combinations that can not live or coexist together) the end of file is indicated by a line that contains two zeros separated by a single space (0 0).
- **Output:**
  OUTPUT DATA (file OUTPUT.TXT) Output is written in the file OUTPUT.TXT as follows:
  - on the first line, two integers separated by space, that represent the maximum number of fish types that can live in the aquarium and the maximum amount of money that can be spent, for this number of fish types.
  - on subsequent lines, the identification numbers of the fish types that can live in the aquarium are given.
- **Example:**
  - *Input*
    170 7
    1 70
    2 50
    3 30
    4 40
    5 40
    6 30
    7 20
    1 4
    1 7
    3 4
    3 5
    5 7
    6 7
    0 0
  - *Output*
    4 160
    2
    4
    5
    6
- **Remark**
  In the case of many solutions to the problem, only one is required to be listed. Time-limit for each test: 5 seconds. Maximal score: 30 points

---

4th Balkan Olympiad in Informatics
Nicosia, Cyprus, 19-25 October 1997

Day 1 - Problem 2 (1-2-3-4 MAZE)

We have a rectangle maze with dimensions nxm (where n<=20, m<=20). The top left and the bottom right positions of the maze are marked by 0, and all other maze positions by one of the numbers 1, 2, 3, 4. The aim of the game is to enter the maze at the top left corner and leave it from the bottom right corner by following the shortest path moving only to the right, left, down and up. The constraint that we have to obey in order to follow a path from the entrance to the exit is to move from a 1, to a 2, to a 3, to a 4, to a 1 and so on, in a cyclic way. From the start position you can only move to a 1 position. You can move to the exit position either from the neighboring upper or neighboring left cell of the exit position.

- **Input:**
  INPUT DATA (file INPUT.TXT)
  The first line contains the integers n and m. Each of the next n lines contains m integers representing the maze.

- **Output:**
  OUTPUT DATA (file OUTPUT.TXT)
  This file will consist of two lines, containing the number of steps and a string of characters respectively. The string consists of the characters D (for down), U (for up), L (for left) and R (for right), the steps along the shortest path to the exit.

- **Example**
  Input
  5 4
  0 1 2 3
  3 2 1 4
  4 1 2 1
  1 4 3 2
  2 3 4 0
  Output
  7
  RRRDDDD

- **Remark:** you may assume that there is always a solution to the problem.

Time-limit for each test: 5 seconds.
Maximal score: 30 points

4th Balkan Olympiad in Informatics
Nicosia, Cyprus, 19-25 October 1997

Day 1 - Problem 3 (Heads and tails)

A NxN matrix (where N<=10) of coins is given, some heads up and some tails up. A player must flip coins so as to achieve an all tails configuration. However, when a player flips a particular coin, each rectilinearly adjacent neighbors (i.e. its left, right, top and bottom neighbors, if they exist), also flip over.
The problem is, given an initial configuration, to achieve an all tails matrix by flipping the minimum number of coins. Note that by flip we mean "point to a coin, and change it and its neighbors".

- **Input:**
  INPUT DATA (file INPUT.TXT)
  The first line contains the integer N. Each of the next N lines contains N characters, H or T standing for the heads/tails respectively.

- **Output:**
  OUTPUT DATA (file OUTPUT.TXT)
  The first line should contain the minimum number of flips. The next N lines comprise a matrix of N integers per line. hese integers have an 1 or a 0 value. The value 1 means that the corresponding coin has been flipped at least once. The alue 0 means that the corresponding coin has not been flipped at all.

- **Examples:**

  - *Input*
    3
    H T T
    H T T
    T T T
  - *Output*
    5
    1 0 0
    0 1 0
    1 1 1

- **Remark** You may assume that there is always a solution to the problem.

Time-limit for each test: 20 seconds.
Maximal score: 40 points
