2001 APICS Programming Contest Problems

The 2001 Atlantic Canada programming Contest was held at St. Francis Xavier University in Antigonish, NS Friday Oct 19 2001. There were 16 teams of three students from nine universities in the four Atlantic provinces.

Each team of 3 students was given 7 problems to try to solve in 5 hours using one computer following the rules for the ACM programming contests.

Each problem has sets of associated data files and matching output files. Files 1 and sometimes 2 are available to the contestants during the competition. The others preceeded with a "J" are reserved for the judges to test submitted solutions.

a. The book shelf problem
b. The common digits problem
c. The tree sum problem
d. The confused lecturer problem
e. The don't take the last one problem
f. The missing digits problem
g. The new villa graph problem

The participating teams came from the folowing universities:

```
University of New Brunswick, Fredericton Campus.
Universite de Moncton , Moncton N.B.
Mount Allison University, Sackville N.B.

Acadia University, Wolfville N.S.
Dalhousie University, Halifax N.S.
St Marys University, Halifax  N.S.
St Francis Xavier University, Antigonish N.S.

University of Prince Edward Island, Charlottetown.

Memorial University of Newfoundland, St. John's.
```

The top three teams from this competition advance to the next level of competition, in Westfield Massachusetts.

Problem A : Book Shelves

Professor Lumber is building a wooden bookshelf with shelves from boards that are glued together in pairs. After gluing, the shelves are sawn to the same width, which is the smallest width of a glued pair. The professor wishes to make the shelves as wide as possible. Write a program that tells the professor how to pair the boards.

The input for the program will consist of an even number of integers, one per line, representing board widths in millimeters.

The output should be the paired board width sums that maximize the minimum pair sum, plus the overall minimum pair sum.

For example:

```
Boards:                                Shelves:

 --------------                         ---------------
|    105       |                       |     105       |
 --------------                         ---------------
                                       |     146       |
 --------------                         ---------------
|    146       |
 --------------

 --------------                         ---------------
|    129       |                       |     115       |
 --------------                         ---------------
                                       |     137       |
 --------------                         ---------------
|    137       |
 --------------

 --------------                         ---------------
|    121       |                       |     121       |
 --------------                         ---------------
                                       |     129       |
 --------------                         ---------------
|    115       |
 --------------
```

Sample input:

```
A1.dat
105
146
129
137
121
115
```

Sample output:

```
A1.out
105 + 146 = 251
115 + 137 = 252
121 + 129 = 250
minimum sum: 250
```

Problem B : Product Digits

Write a program that takes as input two positive integers.
The integers must not be equal and can be input in either order.
Each integer contains at least one and no more than four digits.
Let I denote the interval of integers determined by these two integer values, endpoints included. There must be one line of output for each pair of integers in I that satisfy the criterion in the following paragraph:

The digits in the two input integers must be the same as those in their product, and any such digit must appear the same number of times in the factors (considered together) as it does in the product. Digit order in the factors or the product is irrelevant.

The format and order of the output line(s), when there are such integers in the given range, must be as illustrated in the sample output data.

If there are no such integers in the given range, the following single line of output must be displayed:

No such integers exist in the given range.

Sample input # 1:

```
B1.dat
4000 3600
```

Sample output # 1:

```
B1.out
3656 and 3710 contain the same digits as their product 13563760.
3683 and 3701 contain the same digits as their product 13630783.
3692 and 3710 contain the same digits as their product 13697320.
3692 and 3791 contain the same digits as their product 13996372.
3701 and 3710 contain the same digits as their product 13730710.
3740 and 3941 contain the same digits as their product 14739340.
```

Sample input # 2:

```
B2.dat
987 1234
```

Sample output # 2:

```
B2.out
No such integers exist in the given range.
```

Problem C : Trees

The input for this problem consists of zero or more pairs of lines. The first line of each pair contains a parenthesized representation of a tree structure. A tree with just one integer is represented by the integer alone. The trees corresponding to the first, third and fifth input lines of the sampe input below are shown:

```
          12                   100                 3
         /  |  \                                  / | \
        /   |   \                                /  |  \
       4    2    5                              3   3   2
            |   / \                             |   |   |
            2  3   1                            2   2   3
```

If an entry R has branches to "children" S1 S2 ... Sn (where n >= 1) then it is written in parenthesis (R T1 T2 ... Tn) where Ti is the representation of Si using these same rules.

Notice that this representation does not allow for empty trees. They are not allowed in this problem and the data files do not have any empty trees.

The second line of each pair contains some integers that are to be searched for in the tree from the first line. For each integer on the second line find all paths from the root (shown at the top above) to a leaf (which has no children) where the sum of the values along the path adds up to that integer.

Notice the single spaces in the output and the order of the output lines. For each integer, the paths that add to that value are printed (with "leftmost" first). If a value like 19 has no matching paths it produces no lines of output.

Sample input:
```
C1.dat
(12 4 (2 2) (5 3 1))
16 18 19 20
100
10 100 1000
(3 (3 2) (3 2) (2 3))
7 8 9
```

Sample output:
```
C1.out
16 = 12 + 4
16 = 12 + 2 + 2
18 = 12 + 5 + 1
20 = 12 + 5 + 3
100 = 100
8 = 3 + 3 + 2
8 = 3 + 3 + 2
8 = 3 + 2 + 3
```

Problem D: Doublespeak

Professor Doublespeak has acquired a habit of speaking in a complicated manner. You have discovered, however, that his real point is always said more than once, and is the longest thing that he repeats. These repetitions are a bit tricky to listen for, especially since they can overlap each other. Luckily the repetitions are always close together (so they would be in the same line of his lecture). Spaces, punctuation, and capitalization are also important, and the repetition is of characters, not necessarily words.

To ensure that you do not miss anything that he says, you have made a copy of his lecture, line by line, and need to write a program to extract the longest string of repeated consecutive characters. Each line can be treated independently, and your program should stop when it reaches the end of the file, or a blank line. Print the result from each line (the longest string repeated in the line) on separate lines of output. You can assume that there is exactly one repeated string that is of the longest length, and that each line contains no more than 80 characters.

Sample input:

D1.dat

```
I am telling you now there will be tests and there will be exams.
I usually give a test on questions and a test about exercises.
One time I gave one on question fourteen and on question four hundred five.
So it may happen on Friday, or Monday, and your assignment is due on Friday.
So study, study, study.
```

Sample output:

D1.out

```
 there will be
 a test
 on question four
 on Friday
 study, study
```

Problem E : Don't Pick The Last One

Consider the following whoever-picks-the-last-one-loses game. The game is played on a 5×5 board. Initially every array cell has a piece in it. Two players remove pieces alternatively from the board. The player can remove any number of consecutive pieces in a row or column. For example, in the configuration depicted below where one indicates a piece, the player can either remove one piece (A1, A2, or B1), or remove two pieces (A1 and A2, or A1 and B1) simultaneously. The game ends when one player is forced to take the last piece, and the other player wins the game.

```
  | 1 2 3 4 5
--+----------
A | 1 1 0 0 0
B | 1 0 0 0 0
C | 0 0 0 0 0
D | 0 0 0 0 0
E | 0 0 0 0 0
```

Write a program that evaluates board configurations from this game. The program must output "winning" when there exists a winning move that no matter how the opponent responds, it will force the opponent to take the last piece. Otherwise, the program must output "losing". Note that during the game tree evaluation, if the current configuration has a winning move (one that the opponent cannot beat), then it is not necessary to search any further because the configuration is guaranteed to be winning. This can greatly reduce the game tree search time.

The input consists of 5i + 1 lines. Line 1 gives the number of configurations i to follow. For each configuration, there are 5 lines each containing a combination of five 0's and 1's that represent a row of the board.

Output simply consists of a line for each configuration showing "winning" or "losing".

Sample input:

```
E1.dat
2
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 0 0
0 0 0 0 0
1 1 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Sample output:

```
E1.out
winning
losing
```

Problem F: Character Decoding

This is a test for decoding the values of characters. Assume a numerical expression is encoded in English characters by replacing some digit numbers (from 0 to 9) with English characters.
So this kind of numerical expressions can be expressed in new forms, such as
2BAD = ABE + CD

Write a program to decode the expressions in characters and output the numerical value of characters, according to the following rules and assumptions.

1. All character values are integers between 0 to 9 both inclusive.

2. An expression is represented as a set of items combined with operators. Only the operators +, - and = are used in each expression. And at most 5 items are used in one expression.

3. There is one and only one operator = in each expression, and only one item is in the left-hand side of the operator =.

4. Each item is represented by a combination of capital English characters and digital numbers. The value of the left-most character in each item is not 0.

5. The input data are represented as several lines of numerical expressions. Each line represents an expression independent of the other lines.

6. Output the value of the left-most item in each expression row by row, in the same order as that in the input.

7. If there are multiple solutions, print out the smallest values for each left-most item. If no possible solutions exist, print out a question mark (?) instead.

Sample input:

```
F1.dat
CA = AB + 6C
DDE5 = DEFG - EHI + DDH
A = 0
```

Sample output:

```
F1.out
81
1115
?
```

Problem G : The New Villa

Mr. Black recently bought a villa in the countryside. Only one thing bothers him: although there are light switches in most rooms, the lights they control are often in other rooms than the switches themselves. While his estate agent saw this as a feature, Mr. Black has come to believe that the electricians were a bit absent-minded (to put it mildly) when they connected the switches to the outlets.

One night, Mr. Black came home late. While standing in the hallway, he noted that the lights in all other rooms were switched off. Unfortunately, Mr. Black was afraid of the dark, so he never dared to enter a room that had its lights out and would never switch off the lights of the room he was in.

After some thought, Mr. Black was able to use the incorrectly wired light switches to his advantage. He managed to get to his bedroom and to switch off all lights except for the one in the bedroom.

You are to write a program that, given a description of a villa, determines how to get from the hallway to the bedroom if only the hallway light is initially switched on. You may never enter a dark room, and after the last move, all lights except for the one in the bedroom must be switched off. If there are several paths to the bedroom, you have to find the one which uses the smallest number of steps, where ``move from one room to another'', ``switch on a light'' and ` `switch off a light'' each count as one step.

The input file contains several villa descriptions. Each villa starts with a line containing three integers r, d, and s. r is the number of rooms in the villa, which will be at most 10. d is the number of doors/connections between the rooms and s is the number of light switches in the villa. The rooms are numbered from 1 to r; room number 1 is the hallway, room number r is the bedroom.

This line is followed by d lines containing two integers i and j each, specifying that room i is connected to room j by a door. Then follow s lines containing two integers k and l each, indicating that there is a light switch in room k that controls the light in room l.

A blank line separates the villa description from the next one. The input file ends with a villa having r = d = s = 0, which should not be processed.

For each villa, first output the number of the test case ("Villa #1", "Villa #2", etc.) in a line of its own.

If there is a solution to Mr. Black's problem, output the shortest possible sequence of steps that leads him to his bedroom and only leaves the bedroom light switched on. (Output only one shortest sequence if you find more than one.) Adhere to the output format shown in the sample below.

If there is no solution, output a line containing the statement

```
        The problem cannot be solved.
```

Output a blank line after each test case.

Sample input:

```
G1.dat
3 3 4
1 2
1 3
3 2
1 2
1 3
2 1
3 2

2 1 2
2 1
1 1
1 2

0 0 0
```

Sample output:

```
G1.out
Villa #1
The problem can be solved in 6 steps:
- Switch on light in room 2.
- Switch on light in room 3.
- Move to room 2.
- Switch off light in room 1.
- Move to room 3.
- Switch off light in room 2.

Villa #2
The problem cannot be solved.
```

You may find the following suggestion helpful.
Suppose the rooms are labelled a, b, c, etc.
If there are 10 rooms the last would be j.
When you are in a room the lights in each of the other rooms can be on or off.
Consider the case of 3 rooms a,b,c. The possible states are
a100 (in room a with lights on in a and off in b and c)
a110 (in room a with lights on in a and b and off in c)
...
a111 (in room a with lights on in a and b and c)
b010 (in room b with lights on in b and off in a and c)
...
c001 (in room c with other lights off)

So for 3 rooms the start state is a100 and you want to find the shortest path to c001. For 5 rooms you start in a10000 and want to get to e00001.

The light must always be on the room you are in. So if there are 10 rooms the lights in the other 9 rooms can be on or off leading to 2^9 = 512 differen states for each room. So for 10 rooms the graph could have 5120 verticies.