



# Canadian Mathematics Competition

An activity of The Centre for Education  
in Mathematics and Computing,  
University of Waterloo, Waterloo, Ontario

# *Canadian Computing Competition*

for the  SYBASE® Awards

**Tuesday, March 3, 1998**

## Problem A

### Censor

Input file : censor.in

Output file : censor.out

The Society for Prevention of Profanity on the Internet has observed a growing number of chat lines on the World Wide Web. A chat line allows a Web user to type lines of text which are transmitted to all other users. The Society is concerned about the number of four-letter words being transmitted by these chat lines and has proposed the mandatory use of software to remove all four-letter words from every transmission. Your job is to write the software to do this removal.

The input to your program consists of an integer,  $n$ , on a line by itself, followed by  $n$  lines of text. Each line of text contains words separated by spaces. Each word consists of letters of the alphabet and exactly one space separates adjacent words. There are no spaces before the first word or after the last word on each line. Lines do not exceed 80 characters in length.

The output from your program should consist of the  $n$  lines of text, with each four-letter word replaced by four asterisks. The lines should be separated by one blank line.

#### Sample input

```
2
The quick brown fox jumps over the lazy dog
Now is the time for all good people to come to the aid of the party
```

#### Sample output

```
The quick brown fox jumps **** the **** dog

Now is the **** for all **** people to **** to the aid of the party
```

## Problem B Cross-Number Puzzle

Input file : none

Output file - part 1 : perfect.out

Output file - part 2 : cube.out

Solve the following three problems. Solutions to the first two problems are to be generated using a computer. The solution to the third problem may be done by hand, but must be entered into the grid provided in the information/answer form.

1. Write a program to print the perfect numbers between 1000 and 9999 inclusive. A perfect number is a positive integer which is equal to the sum of its proper divisors. A proper divisor is any divisor less than the number itself. For example, 6 is a perfect number since  $1 + 2 + 3 = 6$ . The numbers should be printed one per line.
2. Write a program to generate all integers between 100 and 999 inclusive which are equal to the sum of the cubes of their digits. The numbers are to be printed one per line.
3. Use the output from parts 1 and 2, and any other programs you care to write, to solve the following cross-number puzzle.

1.	2.		3.
4.		5.	
	6.		

Clues:

ACROSS :

1. A 4-digit perfect number.
4. A 4-digit palindrome. (A palindrome is a number which reads the same right to left as it does left to right. For example : 1221)
6. A 3-digit number that is equal to the sum of the cubes of its digits.

DOWN :

2. A pair of twin primes. (Primes which differ by 2; for example 3 and 5 are twin primes)
3. A 4-digit perfect square.
5. A prime number.

## Problem C Mars Rover

Input file : rover.in

Output file : rover.out

A planetary "rover" is traveling on a perfectly level surface (no obstacles) on Mars. The rover is in radio communication with the "lander" it arrived in. The lander accumulates and relays commands, which it receives from Earth, to the rover. The rover makes several excursions. Each excursion begins with the rover at the lander facing in a known direction. At the end of each excursion the lander must compute and transmit a sequence of instructions to return the rover to the lander.

The rover responds to a sequence of commands sent from the lander. Each command tells the rover to move ahead some multiple of an exact unit of distance, or to turn left or right exactly 90 degrees.

The "move ahead" command is encoded using two consecutive lines in the input file. The first line contains the integer 1, and the second line contains a non-negative integer  $n$ , the distance to move forward.

The "turn right" command is encoded using a single input line containing the integer 2.

The "turn left" command is encoded using a single input line containing the integer 3.

For example, the following sequence of commands instructs the rover to turn left, then move ahead 10 units, then turn right, and then move ahead 5 units.

```
3
1
10
2
1
5
```

Your task is, given such a sequence of commands, to determine how far the rover must travel to return to the lander, and to determine the shortest sequence of commands that will return the rover to the lander. In the example above, the rover must travel 15 units to return, and the shortest sequence of commands is

```
2
1
10
2
1
5
```

The input file begins with a line containing a positive integer,  $n$ , the number of excursions for the rover. The commands for the excursions occupy subsequent lines of the input file. Each excursion consists of a number of commands followed by a line containing 0. There are no errors or blank lines in the input. The rover travels less than 10 000 units of distance on each excursion.

continued..

For each excursion, the output file should contain a line:

```
Distance is k
```

where k is the distance in units that the rover must travel to return to the lander. The following lines should contain the shortest sequence of commands to return the rover to the lander. A blank line should separate the lines of output for different excursions.

Note: Partial marks will be given for the distances without the return trips.

### Sample input

```
3
2
3
3
0
3
1
10
2
1
5
0
1
5
2
1
5
3
3
1
1
0
```

### Sample output

```
Distance is 0

Distance is 15
2
1
10
2
1
5

Distance is 9
1
4
3
1
5
```

## Problem D Lottery

Input file : lottery.in    Output file : lottery.out

You have just won the lottery. All that separates you from your multi-million dollar prize is your correct answer to the following skill-testing question:

$$1234 + 4567 \times 11$$

In your twenty seconds you see your fortune slipping away because you don't know whether the answer is

$$(1234 + 4567) \times 11 = 63811$$

or

$$1234 + (4567 \times 11) = 51471.$$

Finally you guess 63811, but that answer is incorrect. Your math teacher set the question and expected you to remember that multiplication is done before addition. The correct answer is 51471.

Your job is to write a program to insert parentheses into lottery questions such as the above so as to make clear the order of operations.

The input to your program consists of a line containing an integer,  $n$ , followed by  $n$  lines of input. Each of the  $n$  lines contains an expression consisting of integers, and the operators +, -, and X (upper case X) denoting addition, subtraction and multiplication respectively. Adjacent integers are separated by one operator. There is a single space to the left and to the right of each operator and no input line contains more than 80 characters.

Your output should consist of the same  $n$  lines, with a blank line between them, with parentheses inserted in the  $n$  lines so as to indicate the order of operations. Multiplication should be done first, from right to left, and addition and subtraction should then be done from left to right. Spaces surrounding operators should be preserved.

### Sample input

```
3
10 + 20 X 30
1 + 2 + 3 - 4
123 + 456 X 789 - 876
```

### Sample output

```
10 + (20 X 30)

((1 + 2) + 3) - 4

(123 + (456 X 789)) - 876
```

## Problem E

### Mountain Passage

Input file : passage.in Output file : passage.out

Alp the mountain climber is on the northwest corner of a square area of mountainous terrain and wishes to find a passage to the opposite (southeast) corner. Alp is currently at an elevation at which oxygen is not needed, but at any higher elevation oxygen is required. Oxygen, when required, is used at the rate of one unit per horizontal step.

The northwest corner of the terrain is at position  $(1, 1)$  and the southeast corner is at position  $(n, n)$ , where  $n$  is a positive integer read from the input file. The elevation of each point  $(x, y)$ ,  $(1 \leq x, y \leq n)$ , is an integer read from the input; each elevation occupies a single line in the input file.

Alp moves in a series of horizontal steps, where each step moves Alp a unit north, a unit south, a unit east, or a unit west. Alp must remain in the square region and cannot climb or descend more than 2 units of elevation in a single step. If the elevation at either the beginning or the end of the step requires oxygen, a unit of oxygen is consumed by Alp during the step.

The first line of input is a positive integer indicating the number of trips that Alp must make. For each trip there is a number of input lines. The first line for each trip contains an integer  $n \leq 25$ , indicating the size of the square area of terrain. The next  $n^2$  lines each contain a single integer indicating the elevation of a particular point of terrain. The elevations are given for the points in the following order:

$(1, 1), (1, 2), (1, 3) \dots (1, n), (2, 1), (2, 2), \dots (n, 1), (n, 2), \dots (n, n)$ .

For each trip, if a passage exists, the output is a single line containing an integer indicating the number of units of oxygen consumed. If a passage does not exist, the output is a single line containing the message "CANNOT MAKE THE TRIP". Output lines for the trips should be separated by a single blank line.

continued...

### Sample input

2  
5  
5  
4  
3  
2  
1  
7  
5  
6  
6  
6  
8  
8  
8  
9  
6  
9  
6  
9  
9  
9  
6  
4  
5  
4  
5  
3  
2  
4  
9  
9  
4

### Sample output

5

CANNOT MAKE THE TRIP