

# ACM Pacific NW Region Programming Contest

## 15 November 2003

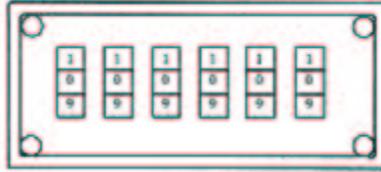
### Problem B: SpinLock

Simulate a locked spinner puzzle.

#### DESCRIPTION

A locked spinner puzzle is a puzzle where you can only change wheels in groups. It is a common puzzle to achieve some value on the spinners with by only changing them in the allowed groups.

Assume a row of  $D$  numbered wheels, such as what is found on a combination lock on a briefcase.



(example: wheels of a briefcase lock)

Each wheel will be labeled sequentially with the digits 0 through 9. The initial state of the wheels is part of the problem specification.

Below this are a series of  $B$  buttons with labels that are  $D$  digits long. For example,  $D$  may be 4 and the labels are 1000 1200 1002 0111 and 0100. Pressing the button labeled 1000 moves the first wheel once, but leaves the others alone, while pressing the button labeled 1002 moves the first wheel once and the fourth wheel twice, leaving the center button unchanged.

Your task is to simulate such a locked spinner puzzle, providing the final readout of the wheels.

**INPUT:** Read from file [b.in](#)

The input to your program will be several spinner puzzles. Each puzzle will be  $D$  digits (at most 10) representing the starting positions of the wheels. Following this, each line will have the button label for which button is pressed next. The end of each spinner puzzle will be marked with the character 'x' alone on a single line. End of the input file will be marked with the character 'z'.

#### Sample Input:

```
0001
1003
0206
0034
1111
1003
x
91234567
12340051
33402401
x
z
```

**OUTPUT:**

# ACM Pacific NW Region Programming Contest

## 15 November 2003

The output file should print the final positions of each spinner puzzle's wheels. Provide the output header "Spinlock Results" on the first line. Each puzzle follows on a separate line, formatted as below:

**Sample Output:**

```
Spinlock Results  
3348  
36976919
```