

# PSH: Safe Cracker

In this problem you are to write a program that assists a safecracker in opening a digital lock. The digital lock has an  $n$ -digit code ( $0 < n \leq 7$ ). The interface to the lock with a 4-digit code looks as in Figure 1. For each slot in the lock there are two buttons for scrolling the digit up or down. Each press of a button will scroll the corresponding digit up or down by one. Through inside information, our safecracker knows what the secret code is. We call this secret code the target code. She also knows a list of forbidden  $n$ -digit codes that she should avoid while manipulating the lock. If the lock digits are set to one of these forbidden codes an alarm will sound, which is not a good thing for safecrackers. Your task is to write a program that will figure out the fastest way for our safecracker to get from the current combination on the lock, called the starting code, to the  $n$ -digit target code without going through any of the forbidden codes.

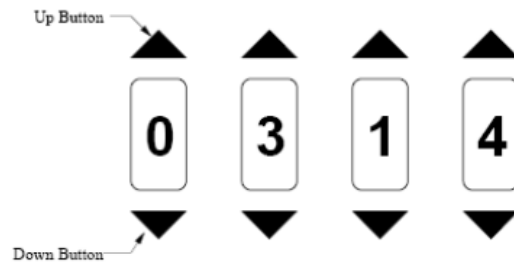


Figure 1. The Digital Lock with  $n = 4$

## Input Format

The program takes, via standard input, the following information:

On the first line is the integer  $n$  specifying the number of digits in the lock.

On the next line is the  $n$ -digit starting code.

On the third line is the  $n$ -digit target code.

On the fourth line is an integer  $f$ ,  $0 \leq f \leq 10^n$ .

On the following  $f$  lines are the forbidden codes, specified one per line.

## Output Format

The program must write to standard output. The format is as follows.

If it is not possible to reach the target code from the starting code without using a forbidden code, the program writes a  $-1$  to the standard output and terminates. Note this should also be the output if the starting code or target code is a forbidden code.

Otherwise, the program writes to the standard output a number  $k$ , representing the minimum number of steps to get from the starting code to the target code without using a forbidden code. At each step one of the buttons is pushed. Each of the next  $k$  output lines shows the  $k$  steps and the result of each step. Each line has the following format.

```
mB code
```

where  $B$  is taken from the set  $\{1, 2, \dots, n\}$ , and represents the digit position with 1 being the leftmost,  $m$  is taken from the set  $\{U, D\}$ , and represents the button, up (U) or down (D), and `code` represents the resulting code after the indicated button is pushed.

Note that if a digit is currently 0, pushing the down button will bring up 9. Similarly, if a digit is currently 9, pushing the up button will bring up 0.

Furthermore, if there is more than one sequence of steps with which you can reach the target, you should output the one that is lexicographically smallest.

## Sample Input

```
4
5555
7777
4
6555
5655
5565
5556
```

## Sample Output

```
10
D1 4555
U2 4655
U2 4755
U1 5755
U1 6755
U1 7755
U3 7765
U3 7775
U4 7776
U4 7777
```

## Explanation

There are multiple sequences of steps that can move us from our start state to our target state in 10 steps. Here are a few of them:

```
D2,U1,U3,U4,U3,U4,U1,U2,U2,U2
D2,U1,U3,U4,U1,U4,U3,U2,U2,U2
D3,U1,U2,U4,U1,U4,U2,U3,U3,U3
```

Out of all of the possible sequences of length 10 that crack the safe, the one that comes first in alphabetical order is the one the first one specified in the list above, so this is the one that we use in the output.