PROB2: SHAKE

Encrypt secret messages.

DESCRIPTION

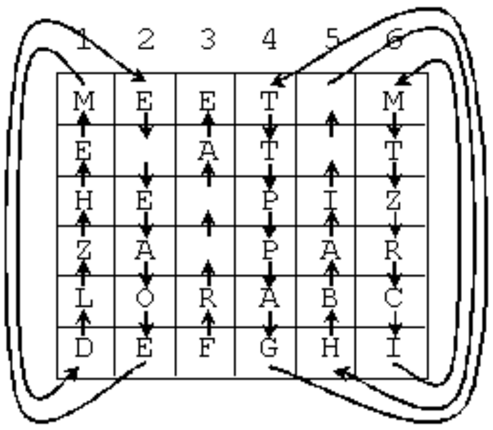
Sam wants to send secret messages to Sally, and visa versa, so they devise a simple but effective encryption scheme they can perform by hand. However, they soon begin to fall in love and their messages to each other are growing longer and longer. Therefore, they decide their encryption scheme needs to be automated. Your job is to write a computer program that will implement their "shake, rattle and roll" encryption scheme. (Decryption will not be implemented at this time.)

A text message is placed into a 2D array in row major order, with each character in a unique cell of the array. If the message does not totally fill the array, the empty cells are filled with the capital letters of the alphabet starting with the letter A and going thru the letter Z (repeated as needed). For example, the message, "Meet me at the pizza parlor" in a 6 by 6 array would look like the figure below. Note that all characters are stored as capital letters.

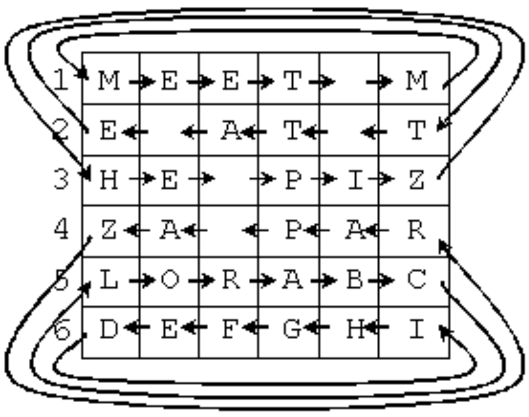
M	E	E	T		M
E		A	T		T
H	E		P	I	Z
Z	A		P	A	R
L	O	R	A	B	C
D	E	F	G	H	I

To encrypt this message, 3 separate operations are performed as follows:

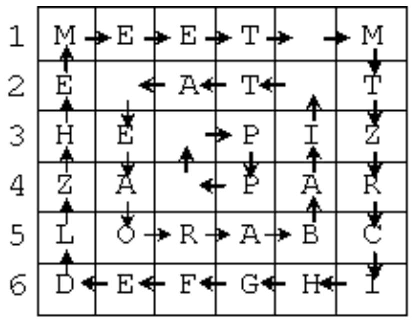
Shake: Each odd column is shifted up one character, with the top most character moving to the bottom of the column. Each even column is shifted down, with the bottom most chararacter moving to the top of the column. The columns are numbered starting at 1. For example:



Rattle: Each odd row is shifted to the right one character, with the rightmost character moving to the leftmost column in the same row. Each even row is shifted to the left one character, with the leftmost character moving to the rightmost column in the same row. The rows are numbered starting at 1 from the top. For example:



Roll: Each odd "loop" around the matrix is rotated to the right one character, while each even "loop" is rotated to the left one character, as shown in the figure below. "Loops" are even or odd based on the row number of their top most row (with the top row being row 1).



The matrix size is specified in the encryption key and can vary in size from 3x3 to 100x100. The matrix is always square.

INPUT: prob2.dat

The input file will contain one or more encryption problems. Each problem takes up two lines of the file. The first line of a problem is the encryption key. The second line is the text to be encrypted. An encryption key always begins with a two digit matrix size, followed by a series of 'S', 'R', or L' characters in any order. For each 'S' character, a "shake" operation is performed, for each 'R' character a 'rattle" operation is performed, and for each 'L' character a "roll" operation is performed, in the order specified. A size of "00" is interpreted as 100.

The encryption key is limited to 80 characters. The message is limited to 10,000 characters. Assume that the message will always fit inside the specified matrix.

An example input file would be

```
       column   111111111122222222223
       123456789012345678901234567890
line 1:04RSRR[EOL]
     2:I love ice cream[EOL]
     3:06SRL[EOL]
     4:Meet me at the Pizza Parlor[EOL]
```

```
      :[EOF]
```

OUTPUT: `prob2.out`

Other than the standard header and trailer messages, the encrypted text is output for each problem. The length of the encrypted text is the size of the matrix squared (e.g., a 3x3 matrix produces a string of length 9).

The correct output corresponding to the example input file would be

```
       column   111111111122222222223333333334
       1234567890123456789012345678901234567890
line 1:Program 3 by team 0[EOL]
     2:IREAELCIMVE   OC[EOL]
     3:EIEEAGTTIMT E P ZHRZB PAORDAFLEA CMH[EOL]
     4:End of program 3 by team 0[EOL]
      :[EOF]
```