Practice S08P08: InfoSec and Cryptography II: Transposition

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 9

Objective: 2D array, Characters and Strings

Task statement:

As more and more information is stored in computers and transmitted over network, it is critical to prevent unauthorized parties from accessing such information.

A common approach for this purpose is to encrypt such information into ciphertext so that it cannot be easily understood by unauthorized parties. In contrast, the authorized parties are informed of how to decrypt the cipher text. Therefore, they will be able to access the information without any problem.

For example, (a simplified version of) an encryption algorithm called the transposition algorithm consists of two steps:

1. Given a message and the dimensions of a 2D array, the letters in the message are placed into the array row by row, top to bottom. If there are unused slots in the array, those slots are filled with a '@' character.
2. Afterwards, all characters are read from the array column by column, left to right, forming the ciphertext.

	[0]	[1]	[2]
arr[0]	G	o	o
arr[1]	d	T	i
arr[2]	m	e	@

Figure 1. After placing the message "GoodTime" into a 3x3 2D array **arr**.

For example, let say the given message is "GoodTime" and the dimensions of the 2D array are 3 x 3 (3 rows and 3 columns). The 8 letters in the message are placed row by row, top to bottom, into the 2D array **arr** as shown in Fig. 1:

- 'G', 'o' and 'o' are in the row 0, column 0 to2.
- 'd', 'T' and 'i' are in row 1, column 0 to 2.
- 'm' and 'e' are in row 2, column 0 to 1.
- The unused slot in **arr** (*i.e.*, arr[2][2]) is filled with '@'.

The characters in **arr** then are read column by column, left to right:

- 'G', 'd' and 'm' are read from column 0.
- 'o', 'T' and 'e' are read from column 1.
- 'o', 'i' and '@' are read from column 2.

Therefore, the ciphertext is GdmoToi@.

To decrypt a message, the two steps are reversed:

1. The characters in the ciphertext are placed into a 2D array of the same dimensions as the one used in encryption, column by column, left to right.
2. The characters are read row by row, top to bottom, with all '@' characters at the end ignored. In this way, the original message will be formed again.

For example, if the given ciphertext is GdmoToi@ and the given dimensions are 3x3, after step 1, the 2D array in Fig. 1 can be reconstructed. After step 2, the original message "GoodTime" can be formed again.

Write a program **transpose.c** to read in an integer (1 or 2), the dimensions of a 2D array (two integers in [1, 10], and a string (a sequence of letters, as well as '@' characters if the string is used for decryption):

- If the given integer is 1, the program encrypts the string using the transposition algorithm.
- If the given integer is 2, the program decrypts the string by reversing the steps in the transposition algorithm as described above.

Your program should contain two functions with the following headers:

```
int encrypt(int rows, int columns,
            char message[], char result[])
int decrypt(int rows, int columns
            char message[], char result[])
```

where **`rows`** and **`columns`** are the dimensions of the 2D array, **`message`** is the given string, and **`result`** is a char array for storing the result of encrypting / decrypting the message.

Both functions return 1 if the encryption / decryption is successful, otherwise they return 0. (You may want to check out the test cases in the file package to figure out the possible reasons why the algorithm would fail.)

You may assume that the given string contains at most 100 letters.

You may write additional function(s) if necessary. You may use any characters and string function(s) if necessary.

Sample runs

```
Enter 1 for encryption, 2 for decryption: 1
Enter dimensions of a 2D array: 2 4
Enter message: GoodTime
Encrypted message: GToiomde

Enter 1 for encryption, 2 for decryption: 1
Enter dimensions of a 2D array: 3 3
Enter message: GoodTime
Encrypted message: GdmoTeoi@

Enter 1 for encryption, 2 for decryption: 2
Enter dimensions of a 2D array: 3 3
Enter message: GdmoTeoi@
Decrypted message: GoodTime

Enter 1 for encryption, 2 for decryption: 1
Enter dimensions of a 2D array: 2 3
Enter message: GoodTime
Encryption failed.

Enter 1 for encryption, 2 for decryption: 2
Enter dimensions of a 2D array: 5 3
Enter message: Gdm@@oTe@@oi@@
Decryption failed.
```