National Institute of Technology, Calicut

Department of Computer Science and Engineering

CS2094 - Data Structures Lab

Assignment #1 (Advanced)

1. Given two strings s1 and s2, the relation *alike* is defined as follows

S1 and s2 are *alike* if they are of the same length and differ in zero, one or two places only. For example, xyz and xab are *alike*, but xywz and xabw are not *alike*, zab and xa are not *alike*.

Input: X and Y, where X and Y are sequences containing the same number of strings, n. X and Y are separated with a newline character.

Output : A sequence B of 1's and 0's of length n such that the k^{th} element of B is 1 if the k^{th} elements of X and Y were alike, and the k^{th} element of B is 0 if the k^{th} elements of X and Y were not alike.

Constraints:

a. 1<=n<=50
b. Strings are composed of lower case alphabets only
c. String lengths vary from 0 to 6
d. A null string is input as 0 (zero)

Examples:
Input: abc defg 0 pi
deft aefg 0 pig
Output: 0110

Input: bat cat mat
0 bta cat
Output: 001

Try to make your algorithm as efficient as possible in terms of space and time. Generate suitable error messages where appropriate. For example if X and Y differ in the number of strings, a message "input length does not match", could be printed.

2. You are an intern at SUPERSUDOS, which is famous for their challenging SUDOKUs. They own a software package to create Sudoku of different sizes, 4, 9, 16… etc. However, their software being not fully tested yet (which is not at all unusual), they are not perfectly sure of whether all the sudokos that their software generates, are solvable. Hence they hired some of your seniors to write a program to test whether an input Sudoku is solvable, and return a solution to it if it was solvable. The enthusiastic group did come up with a program to do that, but again, not surprisingly, not all the solutions provided by the senior's software was correct, i.e, sometimes the software generated solutions that were not correct solutions to the given input Sudoku. Hence SUPERSUDOS hired you, to write a program which takes as input a Sudoku, and a solution, and prints whether the given solution is a correct solution to the given Sudoku or not. Consequently, you have to design and implement a program to do that, with suitable error messages.

Input: A valid n, where n is the number of rows in the Sudoku, the Sudoku , and a Solution i.e a filled in Sudoku

Output: Yes, if the solution is correct

No, with suitable error message, if it is not correct a correct solution.

Constraint :

a. 4<=n<=100
b. Input sudokos would be logically correct, even though they may be unsolvable. For example an input Sudoku would never have two ones in the same row, but input solution (obviously, wrong) may have that.
c. Blank spaces in the Sudoku is input as a 0(zero)

Samples:

```
Input: 4
Sudoku:
          1 0  3 0
          4 0  2 0
          0 1  0 0
          0 0  0 0
 Candidate solution:
         1 2 3 4
         4 3 2 1
         2 1 4 3
         3 4 1 2

Output: Yes


Input: 4
```

```
 Sudoku:
           1 0 3 0
           4 0 2 0
           0 1 0 0
           0 0 0 0
  Candidate solution:
           1 2 3 4
           4 3 2 1
           2 1 3 4
           3 4 1 2

Output: No, repeated 3 in column 3



  Input: 4
  Sudoku:
           1 0 3 0
           4 0 2 0
           0 1 0 0
           0 0 0 0

  Candidate solution:
           1 2 3 4
           4 3 1 2
           2 1 4 3
           3 4 2 1
Output: No, solution inconsistent with Sudoku, column 3, row 2
```

In case you require additional material on Sudokus, you may

1. Search the web
2. Ask your classmates who solve sudokus
3. Ask your teachers