COMPUTER PROGRAMMING
2016-2017 FALL TERM

HOMEWORK #4
Due Date : 11.11.2016 ; 23:55

/* Question About Homework 4 to Betül Türkkol */
/* Forum on Moodle */

1. Assume the Sudoku program which is covered in the class last week, write a C program contains 3 functions as following:

void init (int S[][9]) : this function will initialize the sudoku array to zero.

int is_legal (int S[][9] , int r, int c, int x) : this function will check if it is legal to put the number **"x"** into the location **(r,c)** and returns **1** if it is legal, **0** if it is not legal.

void place(int S[][9] , int r, int c, int x) : this function will write the number **"x"** into the location **(r,c)**.

You will write 2 more functions as following :

void read_from_file (FILE *file_pointer, int S[][9]) : this function will read the file you are given, fill the Sudoku array and print the array on the screen.

Rules :
- Each line has 9 integers, each integer should be assigned to the corresponding index of array.

Example : The 3rd number on the 2nd line of the file will be assigned to the index (1,2) of the array.
- Before writing the number into the location, check if the number is legal for that location.
- If all numbers legal to write into the given location, fill the array and print on the screen; otherwise print an error.

void write to the file (FILE *file_pointer, int S[][9]) : this function will write a sudoku array into the file. Each line in the file should contain one row of the array.

2. Assume you are given a word puzzle ,with arbitrary size (mXn), that contains some meaningful words right-to-left, left-to-right, top-to-bottom or bottom-to-top, write a C program to find a given word on the puzzle.

Your program will print the whole puzzle and ask the user to give a word, location of the first character and the orientation. After each guess your program should check the puzzle for the word, print if it is found or not on the puzzle and ask to the user if he/she wishes to quit or continue.

Your program should contain a function that checks if the given word is on the puzzle starting the location through the given orientation, and returns **0** if it is not found, **1** if it is found.

int find (char Puzzle[] [Max_N],int row , int column ,char orientation, char *word) : where

- "orientation" is
  - 'r' or 'R' for right-to-left ;
  - 'l' or 'L' for left-to-right ;
  - 't' or 'T' for top-to-bottom ;
  - 'b' or 'B' for bottom-to-top ;
- row and column starts from 0
- Max_N will be defined as a macro

Example Puzzle :

```
char MyPuzzle[][]={
{'I','N','E','U','E','L',G','C'},
{'T','L','E','M','O','N','E','O'},
{'S','C','H','E','R','R','Y','T'}};
```

For example your function will return **1** for **find(MyPuzzle,2,1,"CHERRY")**