|  |
| --- |
| Computer Programming Language |

【Fall, 2015】

Homework 7

**Program A： Streams and File I/O (50%)**

The text file **words.txt**, which is provided on our CEIBA course website, contains an alphabetically sorted list of English words. Note that the words are in mixed upper and lowercase.

Write a program that reads this file and finds the longest word that reverses to a different word. For example, “stun” reverses to make the word “nuts” but is only four letters long. Find the longest such word.

**Program B： Classes (50%)**

Create a class named **Fractions** containing two integer data members named **num** and **denom**, used to store the numerator and denominator of a fraction having the form num/denom. Your class should include a default constructor that initialized **num** and **denom** to 1 if there is no user initialization, and it must prohibit a 0 denominator value. In addition, create member functions for displaying an object’s data value and overloaded operator functions for adding, subtracting, multiplying, and dividing two **Fraction** objects, as follows:

Addition: a/b + c/d = (a \* d + b \* c) / (b \* d)

Subtraction: a/b - c/d = (a \* d - b \* c) / (b \* d)

Multiplication: a/b \* c/d = (a \* c) / (b \* d)

Division: (a/b) / (c/d) = (a \* d ) / (b \* c)

Finally, your class should have a member function that reduces each fraction to its terms as well as input and output function for entering and displaying a fraction. Implement a C++ main program that tests each member function.

**Challenge Program: Sodoku (Bonus Points 50%)**

Solve a partially filled-in normal 9×9 Sudoku grid and display the result in a human-readable format. Backtracking algorithm, but not the only algorithm, is usually used to solve the Sudoku that iterates all the possible solutions for the given sudoku. If the solutions assigned do not lead to the solution of Sudoku, the algorithm discards the solutions and rollbacks to the original solutions and retries again and hence the name backtracking.

Below is the general pseudocode of backtracking algorithm for standard sudoku template (9×9):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Initialize 2D array with 81 empty grids (Row=9, Col= 9)

Fill in some empty grid with the known values

Make an original copy of the array

Start from top left grid (nx = 0, ny = 0), check if grid is empty

if (grid is empty) {

assign the empty grid with values (i)

if (no numbers exists in same rows & same columns same as (i) & 3x3 square (i) is currently in)

fill in the number

if (numbers exists in same rows & same columns same as (i) & 3x3 square (i) is currently in)

discard (i) and repick other values (i++)

}

else {

while (nx < 9) {

Proceed to next row grid(nx++, ny)

if (nx equals 9) {

reset nx = 1

proceed to next column grid(nx,ny++)

if (ny equals 9) {

print solution

}

}

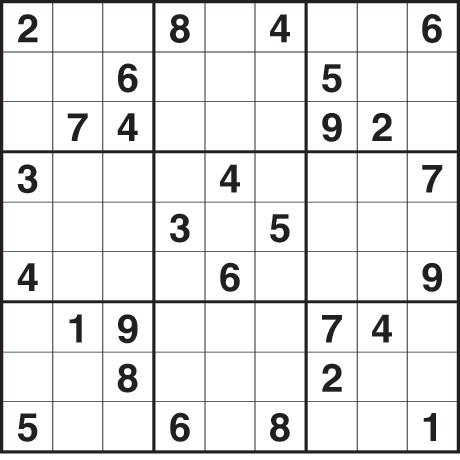
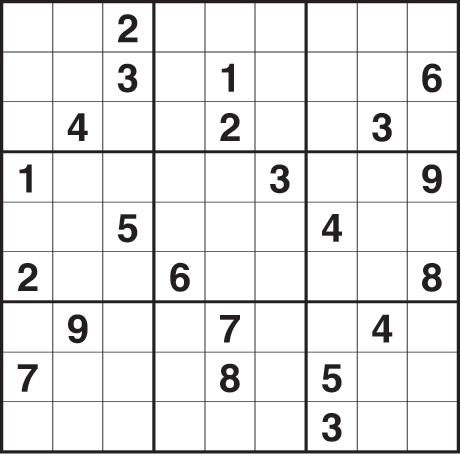
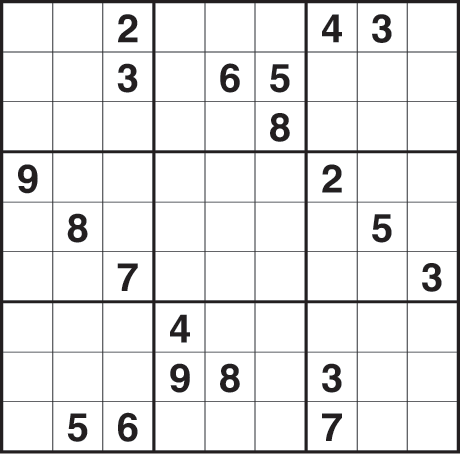
}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Your program should allow the user to input a Sodoku game to be solved via a text file and then save the solution in an output file as well as displaying it on the screen. You also need to record and show the computation time of the solution process.

Test your program with the following puzzles representing three different difficulty levels. Compare the computation time of these three puzzles.

**Notes:**

1. Please submit your programs (source codes and execution files) to the CEIBA course website before **Jan. 7**. Hand in the hardcopies of your program codes in the class of **Jan. 7 (3:30PM)**.
2. Late submission will have a penalty of 10% discount per day of your grade toward a minimum score of 60. No late submission over a week will be accepted.
3. Criteria of grading include: (1) Program functionality; (2). User interface; (3). Structure of the program; (4). Suitable comments; (5). Programming style; (6). Creativity.