Homework (Functions)

CSE 4108

Structured Programming I Lab

November 2022

Homework

1. **Taxing Function!:**

In **Lab 4 - Selection Statements (problem: Tax Season)**, you wrote a program that asks the user to enter the amount of taxable income, then displays the tax due. Modify the program so that it uses a function to compute the amount of income tax. When passed an amount of taxable income, the function will return the tax due.

2. **Random Walking Function!:**

In **Lab Arrays (problem: Random Walk)** you wrote a program that generates a "random walk" across a 10 × 10 array. Modify the program so that it includes the following functions:

```
void generate_random_walk (char walk [10][10]);
void print_array (char walk [10][10]);
```

main first calls generate_random_walk, which initializes the array to contain '.' characters and then replaces some of these characters by the letters A through Z, as described in the original problem. main then calls print_array to display the array on the screen.

3. **Anagrams 2:**

In **Lab Arrays (problem: Anagram)**, you wrote a program that tests whether two words are anagrams. Modify the program so that it includes the following functions:

```
void read_word (int word [26]);
bool equal_array (int word_1 [26] , int word_2 [26]);
```

main will call read_word twice, once for each of the two words entered by the user. As it reads a word, **read_word** will use the letters in the word to update the counts array, as described in the original problem. (main will declare two arrays, one for each word. These arrays are used to track how many times each letter occurs in the words). main will then call **equal_array**, passing it the two arrays. **equal_array** will return true if the elements in the two arrays are identical (indicating that the words are anagrams) and false otherwise.

4. **Magic Square 2**:

In **Lab Arrays (problem: Magic Square)**, you wrote a program that prints an **n×n** magic square.Modify the program so that it includes the following functions:

```
void create_magic_square (int n, char magic_square [99][99]);
void print_magic_square (int n, char magic_square [99][99]);
```

5. **I got a pair' o dice!:**

Write a program that simulates the game of craps, which is played with two dice. On the first roll, the player wins if the sum of the dice is 7 or 11. The player loses if the sum is 2, 3, or 12. Any other roll is called the "point" and the game continues. On each subsequent roll, the player wins if he or she rolls the point again. The player loses by rolling 7. Any other roll is ignored and the game continues. At the end of each game, the program will ask the user whether or not to play again. When the user enters a response other than y or Y, the program will display the number of wins and losses and then terminate.

```
Sample run:
You rolled: 8
Your point is 8
You rolled: 3
You rolled: 10
You rolled: 8
You win!
Play again? y
You rolled: 6
Your point is 6
You rolled: 5
You rolled: 12
You rolled: 3
You rolled: 7
You lose!
Play again? y
You rolled: 11
You win!
Play again? n

Wins: 2 Losses: 1
```

Write a program as three functions: **main**, **roll_dice**, and **play_game**. Here are the prototypes of the latter two functions:

```
int roll_dice ();
bool play_game ();
```

roll_dice should generate two random numbers, each between 1 and 6, and return their sum. **play_game** should play one craps game (calling **roll_dice** to determine the outcome of each dice roll); it will return true if the player wins and false if the player loses. **play_game** is also responsible for displaying messages showing the results of the player's dice rolls. main will call **play_game** repeatedly, keeping track of the number of wins and losses and displaying the "**you win**" and "**you lose**" messages.

Hint: Use the rand function to generate random numbers. See the **deal.c** program for an example of how to call **rand()** and the related **srand()** function.