CIS 190 Introduction to Procedural Programming Fall 2020

Homework 1

Assigned October 1, 2020

DUE October 11, 2020

Disclaimer: This is INDIVIDUAL homework. No collaboration is allowed, neither is working together. Discussions are fine, but writing code to solve the assignment is individual work. If found that your code is very similar to another student, all students with similar code will be assigned 0 points for the assignment without explanations who copied from whom or collaborated with whom! If you "borrow" code from Internet or a person not in the class, you will be assigned 0 points again without discussions.

What to submit: source code running on UNIX, all applicable files used by your source code. No object, executable files are to be submitted. **Choose between Problem 2 and Problem 3! Solving both problems will NOT give you extra credit!** Mind that Problem 3 carries 43 points vs the 40 points that Problem 2 has. Make a decision how to "invest".

Problem 1 (38 points):

Write a C program to prompt the user to enter a number between 1 and 15462. The final goal is to display the individual digits of the numbers on a line with three spaces between the digits. The first line is to start with the left-most digit and print all digits; the second line is to start with the second digit from the left and print remaining digits, and so forth. In order to do that, you must first separate the digits. No conversion from string to integers is allowed, neither are arrays. For example if 1234 is entered, the following should be printed:

```
0   0   0   0   0
    1   1   1   1
        2   2   2
            3   3
                4
```

Then, write C code to accomplish a similar print, where the digits are displayed flush left and in reverse order. Again, the digits must be separated programmatically. No conversion from string to integers is allowed, neither are arrays. For example:

```
4   3   2   1   0
3   2   1   0
2   1   0
1   0
0
```

Grading:

NO arrays!! NO functions!! NO string-to-number conversions!!
Penalty for each infraction is -10 points (each!!!)

Comments	5 points
Functioning code	10 points
Well-written code	5 points

Good user prompts and feedback	5 points
Correct print/output	10 points
Full submission	3 points

Problem 2 (40 points):
Write a program that will generate (random number generator), but not display, a three-digit "target" number. To make matters interesting, two of the digits must be the same, e.g. 272. However, zero is not allowed, e.g. 370 is invalid number for the game. Use random number generator. Then, allow the user to input a maximum of ten user guesses and for each guess, output the number of hits and matches in the guess. Stop when the user guesses the number or runs out of guesses. For example, if the target is 422, the guess 474 has one hit (4) and one match (4).
NOTE: the three-digit number (generated and user input) must be split into digits by using simple integer division or modulo operations! To make matters interesting, two of the digits must be the same, e.g. 272. However, zero is not allowed, e.g. 370 is invalid number for the game.

Sample output:

```
Today we will play a game, you take ten guesses of a 3-digit number, and I
will tell you which one is a match or hit

This is your number 1 guess
What is your guess?
123
Number 2 is a match

This is your number 2 guess
What is your guess?
456
Number 4 is a match


This is your number 3 guess
What is your guess?
789
Number 7 is a match

This is your number 4 guess
What is your guess?
247
Number 2 is a hit
Number 4 is a hit
Number 7 is a hit


Congratulations, you correctly guessed 247
```

Grading:
NO arrays!! NO functions!! NO string-to-number conversions!!
Penalty for each infraction is (-10 points each!!!)

Comments	5 points
Functioning code per requirements	10 points
Well-written code	5 points
Correct game, ordered output	12 points

Good user prompts and feedback	5 points
Full submission	3 points

Problem 3 (43 points):
Write a C program to print a calendar for an entire year. The desired year is read via keyboard. You need to calculate the first day of the year as well as whether the year is leap or not. The leap year calculation is elaborated at http://www.wikihow.com/Calculate-Leap-Years . The first day calculation can be found at http://mathforum.org/library/drmath/view/55837.html .

Here is a sample output for one month:

January						2000
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Grading:
NO arrays!! NO functions!! NO string-to-number conversions!!
Penalty for each infraction is (-10 points each!!!)

Comments	5 points
Functioning code	10 points
Well-written code	5 points
Good user prompts and feedback	5 points
Correct calendar	10 points
Ordered, clean output	5 points
Full submission	3 points