ICS 4U1: Assignment #1

Create a folder named Assignment1 on your school network G: drive and save all projects there. Save each problem in a separate folder. All programs will have a header (name, date, purpose) and have appropriate comments.

1. Write a program that prints a calendar for one month. Input consists of an integer specifying the first day of the month (1 = Sunday) and an integer specifying how many days are in a month. Use the project name A1Q1.

   Sample Output:

   First day of the month: 3
   Number of days in the month: 31

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		

   Create a function: void calendar(int firstDay, int numberOfDays) that prints the above calendar.

2. Write a program that will generate acronyms. The program should accept a sequence of words as input and respond with a single word that is composed of the first letter of each word in the input. Spaces can appear anywhere in the input string. Create a string function that returns the acronym to implement this. Use the project name A1Q2.

   e.g. INPUT: International Business Machines
   Acronym: IBM

3. Write a program that will ask for two fractions (as four integers) and will then print out the sum of those two fractions. Ensure that the results will be printed out in reduced mixed number format. Use the project name A1Q3.

   e.g. ¾ + ½ = 1 ¼ ½ + ½ = 1

4. Crossword puzzles have words that cross on a common letter. You are to write a program that allows two words to cross at that common letter. Your input will consist of two words. Print the two words crossing on the common letter in crossword fashion. The first word entered will be the horizontal word and the second the vertical word. Your program will stop execution when one or both of the words is "q" . Use the project name A1Q4.

```
Across word? nowhere
Down word? madding

m
a
d
d
i
nowhere
g

Across word? quarrel
Down word? banking

  b
quarrel
  n
  k
  i
  n
  g
```

5. This is a board game played on 9 squares with the digits 1 through 8, and a 9th square that contains an asterisk (*). The board is displayed in three rows and three columns. For example, an arrangement might look like this:

```
1 * 3
4 2 6
7 5 8
```

A valid move consists of swapping the special symbol with any of its four neighbours (up, down, left or right) by typing in those four neighbouring numbers. For example, typing 2, then 5 then 8 will result in the winning position:

```
1 2 3
4 5 6
7 8 *
```

The game begins with the squares in a random order, and to get them into the sorted order shown above, using only valid swaps between the asterisk (*) and neighbouring numbers. Only accept valid input (digits 1-8) and valid moves (up, down, left or right squares adjacent to the asterisk). Invalid moves will not change the board. The game ends when you win (in order: 1 2 3 4 5 6 7 8 *) or when you lose (out of order: 1 2 3 4 5 6 8 7 *). Use the project name A1Q5.

Created by Prof. Michelle Craig, University of Toronto.

DUE DATE: Friday February 28, 2020 at the end of class