Introduction

In this workshop, you will code a C language program that implements simple validation on a series of user input values, and then analyse the data to provide a statistical summary.

Topic(s)

• Computations: **Logic** (sequence, selection, iteration, flags, nesting)

Learning Outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities:
• to create a simple interactive program
• to code a decision using a selection construct
• to code repetitive logic using an iteration construct
• to nest a logical block within another logical block
• to describe to your instructor what you have learned in completing this workshop

1. Review the "Part-1 Output Example" (next section) to see how this program is expected to work

2. Code your program in the file named "**w1p1.c**" **IMPORTANT**: Do NOT use **arrays** in this workshop!

3. After the system library **#include**, and before the **main** function, define two (2) macros:

```
#define MIN_YEAR 2010
#define MAX_YEAR 2021
```

4. Inside the main function, **declare** two (2) **unmodifiable** integer variables "**JAN**" and "**DEC**" representing the first and last months of the year respectively (**initialize** "JAN" to **1** and "DEC" to **12**)

5. Display the title for the well-being log application

6. **Nest** inside an **iteration** construct the following:

    a) Display the following message:

```
        >Set the year and month for the well-being log (YYYY MM): <
```

    b) **Read** from standard input (keyboard) the **year** and **month** (entered on the same line with a space between) assigning the input values to two **integer** variables (having **meaningful names** representing the data they store)

    c) Apply what you have learned about **selection** to define the necessary logic that will validate the values entered for the year and month.

    - The entered year value must be between **MIN_YEAR** and **MAX_YEAR** inclusive
    - The entered month value must be between **JAN** and **DEC** inclusive
    - If any of the above validations fail, the respective error message(s) should be displayed (see example output to see what each error message should display)

7. Step #6 should continue to iterate until a valid year and month value is entered

8. When a valid year and month is entered, display a message indicating the log starting date has been successfully set:

```
>*** Log date set! ***<
```

9. Display the log start date in the format: **YYYY-MMM-DD**
YYYY: The year as 4-digits
MMM: First 3-characters of the month name
DD: The 2-digit day
Note: The log will start on the 1st day of the month entered by the user
Hint: You need to implement alternative/multiple selection to map the month integer value to the respective 3-character month representation. There are a couple of constructs available to you that will make this possible!

Part-1 Output Example *(Note: Use this data for submission)*

```
General Well-being Log
======================
Set the year and month for the well-being log (YYYY MM): 2009 1
      ERROR: The year must be between 2010 and 2021 inclusive
Set the year and month for the well-being log (YYYY MM): 2022 1
      ERROR: The year must be between 2010 and 2021 inclusive
Set the year and month for the well-being log (YYYY MM): 2021 0
      ERROR: Jan.(1) - Dec.(12)
Set the year and month for the well-being log (YYYY MM): 2021 13
      ERROR: Jan.(1) - Dec.(12)
Set the year and month for the well-being log (YYYY MM): 2009 0
      ERROR: The year must be between 2010 and 2021 inclusive ERROR: Jan.(1) - Dec.(12)
Set the year and month for the well-being log (YYYY MM): 2022 13
      ERROR: The year must be between 2010 and 2021 inclusive ERROR: Jan.(1) - Dec.(12)
Set the year and month for the well-being log (YYYY MM): 2021 2
*** Log date set! ***

Log starting date: 2021-FEB-01
```