Workshop 6

Worth: 2.25% of final grade

Breakdown

- Part-1 Coding: 10%
- Part-2 Coding: 40%
- Part-2 Reflection: 50%

Submission Policy

- Part-1 is due **1-day** after your scheduled LAB class by the **end of day 23:59** EST (UTC – 5)
- Part-2 is due **5-days** after your scheduled LAB class by the **end of day 23:59** EST (UTC – 5)
- Source (.c) and text (.txt) files that are provided with the workshop MUST be used or your work will not be accepted. Resubmission will be required attracting a **15% deduction**
- **Late submissions will NOT be accepted**
- **All work must be submitted by the matrix submitter – no exceptions**
- **Reflections will not be read or graded** until the coding parts are deemed acceptable and graded.
- **All files** you create or modify MUST contain the following declaration at the top of all documents:

```
/*****************************************************************************
            <assessment name example: Workshop - #6 (Part-1)>
Full Name  :
Student ID#:
Email      :
Section    :

Authenticity Declaration:
I declare this submission is the result of my own work and has not been
shared with any other student or 3rd party content provider. This submitted
piece of work is entirely of my own creation.
*****************************************************************************/
```

Notes

- Due dates are in effect **even during a holiday**
- You are responsible for **backing up your work regularly**
- It is expected and assumed that for each workshop, you will plan your coding solution by using the computational thinking approach to problem solving and that **you will code your solution based on your defined pseudo code algorithm**.

Late Submission/Incomplete Penalties

If any Part-1, Part-2, or Reflection portions are missing, the mark will be **ZERO**.

Introduction

In this workshop, you will code and execute a C language program that implements a simple validation on a series of user input values that are stored to arrays and later analyzed to produce a variety of summary reports. The program will ask for the user's monthly income and then ask for the price and priority of a series of items the user would like to purchase in the future. It will store this information and allow the user to view predictions on how long it will take to save enough money to purchase their wish list items.

Topic(s)

- Arrays

Learning Outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities to:

- to store data of common/primitive type using an array structure
- to associate related data using parallel arrays
- to process the elements of an array using an iteration construct
- To describe to your instructor what you have learned in completing this workshop

Part-1 (10%)

Instructions

Download or clone workshop 6 (**WS06**) from https://github.com/Seneca-144100/IPC-Workshops
Note: If you use the download option, make sure you **EXTRACT** the files from the .zip archive file

1. Carefully review the "Part-1 Output Example" (next section) to see how this program is expected to work
2. Code your solution to Part-1 in the provided "**w6p1.c**" source code file.
3. Begin by prompting the user for their **NET monthly** income
   - The monthly income must be at least **$500.00**, and not more than **$400,000.00**
   - The minimum and maximum values should be stored in **unmodifiable variables** and used in the validation logic accordingly
   - Display an appropriate error message if the entered value is outside this range
   - **Validation** must be **nested in an iteration construct** and repeat until a valid value is entered
4. Next, prompt the user to specify the **number of wish list items** they want to use in the forecast
   Note
   - The maximum number of items should be limited to **10** (define a macro to help with this)
   - Display an appropriate error message if the entered value is outside this range
   - **Validation** must be **nested in an iteration construct** and repeat until a valid value is entered
5. Now you are ready to **store the wish list item details**. Use an iteration construct to Iterate the number of times necessary to obtain the number of wish list item details specified by the user (from step #4)
6. The item details are made-up of **three (3) related pieces of information** and must be stored in **matching (parallel) arrays**:
   a) Cost
      - A **double** floating-point value representing the value of the item
      - The entered value must be at least **$100.00** (use an **unmodifiable variable** to help with the validation logic accordingly)
      - Display an appropriate error message if the entered value is invalid
      - **Validation** must be **nested in an iteration construct** repeating until a valid value is entered

b) Priority
   - An **integer** value representing the priority of the item
   - The entered value must be **between 1 and 3 inclusive** where:
     - 1 = a must-have item
     - 2 = important to have item
     - 3 = want to have item
   - Display an appropriate error message if the entered value is out of range
   - **Validation** must be **nested in an iteration construct** repeating until a valid value is entered

c) Finance Options
   - A **character** value representing if an item has financing options (don't need to pay entire value up-front)
   - The entered value can only be a lowercase **y** or **n**
   - Display an appropriate error message if the entered value is not a **y** or **n**
   - **Validation** must be **nested in an iteration construct** repeating until a valid value is entered

7. After storing the data to **parallel array's**, display a formatted table of the data entered
   - Use the following printf statements for the **table header**:
   ```
   printf("Item Priority Financed        Cost\n");
   printf("---- -------- -------- -----------\n");
   ```
   - Use the following printf formatting to display **each wish list item** record:
   ```
   printf("%3d  %5d    %5c    %11.2lf\n", ...
   ```
8. After all the data is displayed, **summarize** it with the **total of all the item costs**. Use the following printf statement to properly align it with the appropriate Cost column:
   ```
   printf("---- -------- -------- -----------\n");
   printf("                     $%11.2lf\n\n", ...
   ```
9. Finally, before ending the application, display an exit message

Part-1 Output Example (Note: Use the YELLOW highlighted user-input data for submission)

```
+--------------------------+
+   Wish List Forecaster   |
+--------------------------+

Enter your monthly NET income: $0
ERROR: You must have a consistent monthly income of at least $500.00

Enter your monthly NET income: $500000
ERROR: Liar! I'll believe you if you enter a value no more than $400000.00

Enter your monthly NET income: $6500.50

How many wish list items do you want to forecast?: 0
ERROR: List is restricted to between 1 and 10 items.

How many wish list items do you want to forecast?: 11
ERROR: List is restricted to between 1 and 10 items.

How many wish list items do you want to forecast?: 3
```

```
Item-1 Details:
   Item cost: $39030.15
   How important is it to you? [1=must have, 2=important, 3=want]: 0
      ERROR: Value must be between 1 and 3
   How important is it to you? [1=must have, 2=important, 3=want]: 4
      ERROR: Value must be between 1 and 3
   How important is it to you? [1=must have, 2=important, 3=want]: 1
   Does this item have financing options? [y/n]: N
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: Y
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: k
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: n

Item-2 Details:
   Item cost: $99.99
      ERROR: Cost must be at least $100.00
   Item cost: $1200000
   How important is it to you? [1=must have, 2=important, 3=want]: 3
   Does this item have financing options? [y/n]: y

Item-3 Details:
   Item cost: $350500.25
   How important is it to you? [1=must have, 2=important, 3=want]: 2
   Does this item have financing options? [y/n]: n

Item Priority Financed        Cost
---- -------- -------- -----------
  1        1        n       39030.15
  2        3        y     1200000.00
  3        2        n      350500.25
---- -------- -------- -----------
                      $ 1589530.40

Best of luck in all your future endeavours!
```

Part-1 Submission

1. Upload (file transfer) your source file "**w6p1.c**" to your matrix account
2. Login to matrix in an SSH terminal and change directory to where you placed your workshop source code.
3. Manually compile and run your program to make sure everything works properly:

   ```
   gcc -Wall w6p1.c -o w6 <ENTER>
   ```

   If there are no errors/warnings generated, execute it: ***w6*** *<ENTER>*
4. Run the submission command below (replace **profname.proflastname** with **your professors** Seneca userid and replace **NAA** with your section):

   ```
   ~profName.proflastname/submit 144w6/NAA_p1 <ENTER>
   ```
5. Follow the on-screen submission instructions

Part-2 (40%)

Instructions

Code your solution to Part-2 in the provided "**w6p2.c**" source code file. Upgrade the solution to Part-1 to include an analysis of the entered data and provide the forecasted number of years and months it will take to save enough to purchase the wish list items.

1. Review the "Part-2 Output Example" (next section) to see how the program is expected to work
2. Display a menu with three (3) options:
   1. All Items (no filter)
   2. By priority
   0. Quit/Exit

   Note:
   - Prompt for a menu selection; where valid values are from **0** to **2**
   - The menu should be in an **iteration construct** and only exit / end the program when **0** is entered by the user
3. If **0** is entered, the program should display the **exit message** and **end**
   - **DO NOT** use spaghetti code tactics by forcing the iteration to jump out of the iteration using statements like ***break, exit(), or goto*** ***(this style of programming is prohibited in this course and will receive ZERO grade)***
   - Use a **control variable (flag)** to control the flow
4. If an **invalid value** is entered (that is not a 1, 2, or 0), then display an appropriate error message and continue to **iterate and prompt for a valid menu selection**
5. When option 1 is entered, **iterate** all wish list items and:
   - **Accumulate** (total) each **item cost**
   - Check if the item has **financing options** (value will be '**y**') and **make note if it** (this will be used later to show an additional "note" in the summary output)
6. When option 2 is entered:
   - This will follow the same directions as described in #5 only you will not accumulate (total) all the items, but will only consider items that **match** on the user entered **priority** value
   - Therefore, before iterating, you must prompt the user to specify a **priority** level to filter by (valid values are between **1** to **3** inclusive)
     - Display an appropriate error message if the entered value is out of range
     - **Validation** must be **nested in an iteration construct** repeating until a valid value is entered
   - Just as described in #5, accumulate the item cost and check for financing options (**only for the items that match on the specified priority level**)

7. After menu options 1 or 2, **display** a **forecast summary**:

- The summary should be wrapped (first and last line) with a double line. Use the following:
```
printf("=================================================\n");
```
- **Display** the appropriate filter used to generate the results (based on option 1 **or** option 2):
```
printf("Filter:   All items\n");           // [option-1]
printf("Filter:   by priority (%d)\n"...   // [option-2]
```
- **Display** the total cost of the items (derived from the filtering option selected)
```
printf("Amount:   $%1.2lf\n", ...
```
- **Display** the forecasted number of years and months it will take to save enough to purchase the items. Hint: The **modulus operator** will help you greatly with this!
- **Display** an extra "Note" only if any of the items had **financial options** to indicate that a shorter time is likely possible

Part-2 Output Example (Note: Use the YELLOW highlighted user-input data for submission)

```
+--------------------------+
+   Wish List Forecaster   |
+--------------------------+

Enter your monthly NET income: $0
ERROR: You must have a consistent monthly income of at least $500.00

Enter your monthly NET income: $500000
ERROR: Liar! I'll believe you if you enter a value no more than $400000.00

Enter your monthly NET income: $6225.88

How many wish list items do you want to forecast?: 0
ERROR: List is restricted to between 1 and 10 items.

How many wish list items do you want to forecast?: 11
ERROR: List is restricted to between 1 and 10 items.

How many wish list items do you want to forecast?: 5

Item-1 Details:
   Item cost: $39030.15
   How important is it to you? [1=must have, 2=important, 3=want]: 0
      ERROR: Value must be between 1 and 3
   How important is it to you? [1=must have, 2=important, 3=want]: 4
      ERROR: Value must be between 1 and 3
   How important is it to you? [1=must have, 2=important, 3=want]: 1
   Does this item have financing options? [y/n]: N
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: Y
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: k
      ERROR: Must be a lowercase 'y' or 'n'
   Does this item have financing options? [y/n]: n
```

```
Item-2 Details:
   Item cost: $99.99
      ERROR: Cost must be at least $100.00
   Item cost: $1200000
   How important is it to you? [1=must have, 2=important, 3=want]: 3
   Does this item have financing options? [y/n]: y

Item-3 Details:
   Item cost: $350500.25
   How important is it to you? [1=must have, 2=important, 3=want]: 2
   Does this item have financing options? [y/n]: n

Item-4 Details:
   Item cost: $15500.75
   How important is it to you? [1=must have, 2=important, 3=want]: 1
   Does this item have financing options? [y/n]: y

Item-5 Details:
   Item cost: $6575.55
   How important is it to you? [1=must have, 2=important, 3=want]: 3
   Does this item have financing options? [y/n]: n

Item Priority Financed        Cost
---- -------- -------- -----------
  1        1        n       39030.15
  2        3        y     1200000.00
  3        2        n      350500.25
  4        1        y       15500.75
  5        3        n        6575.55
---- -------- -------- -----------
                     $ 1611606.70

How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 3

ERROR: Invalid menu selection.

How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 1

=================================================
Filter:   All items
Amount:   $1611606.70
Forecast: 21 years, 7 months
NOTE: Financing options are available on some items.
      You can likely reduce the estimated months.
=================================================
```

use month to cal, not year, if months any leftover then up to year.

optional , with fiances y

one flag in here for NOTE

```
How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 2

What priority do you want to filter by? [1-3]: 1
```

loop, not coding than one

```
=====================================================
Filter:   by priority (1)
Amount:   $54530.90
Forecast: 0 years, 9 months
NOTE: Financing options are available on some items.
      You can likely reduce the estimated months.
=====================================================

How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 2

What priority do you want to filter by? [1-3]: 2

=====================================================
Filter:   by priority (2)
Amount:   $350500.25
Forecast: 4 years, 9 months
```

optional , without fiances n

```
=====================================================

How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 2

What priority do you want to filter by? [1-3]: 3

=====================================================
Filter:   by priority (3)
Amount:   $1206575.55
Forecast: 16 years, 2 months
NOTE: Financing options are available on some items.
      You can likely reduce the estimated months.
=====================================================

How do you want to forecast your wish list?
 1. All items (no filter)
 2. By priority
 0. Quit/Exit
Selection: 0

Best of luck in all your future endeavours!
```

Reflection (50%)

Instructions

Record your answer(s) to the reflection question(s) in the provided "**reflect.txt**" text file

1. Why are there three types of iteration constructs if we can make any one of them behave the same way? Refer to your code in this workshop to help backup your answer (do NOT include the code in your answer). Typically, when working with arrays, there is one iteration construct we favour – what is it and explain why it is more preferred over the other options.

2. Describe what you did to both **test** and **debug** your program. How did you go about finding where the problems were located (explain how you did this for both **syntactic** and **semantic** problems)?

3. When attempting to submit your work, you receive the following error feedback from the submitter:

```
Checking output:
In line number 6 of your output:
The output should be:
Enter your monthly NET income: $0
                               ^
But your output is:
Enter your monthly NET income: $567.89
                               ^

Unmatched character details:
The character in column 33 is supposed to be:
  [0] ASCII code(48)
but you printed
  [5] ASCII code(53)

Outputs don't match. Submission aborted!

To see exactly what is wrong, open the following two files in this
directory and compare them:
Your output file:    output.txt
Correct output file: w6p2_master_output.txt
```

Using the above explicit information, **explain each step** you must take to successfully identify and fix your error.

Academic Integrity

It is a violation of academic policy to copy content from the course notes or any other published source (including websites, work from another student, or sharing your work with others).

Failure to adhere to this policy will result in the filing of a violation report to the Academic Integrity Committee.

Part-2 Submission

1. Upload your source file "**w6p2.c**" to your matrix account
2. Upload your reflection file "**reflect.txt**" to your matrix account (to the same directory)
3. Login to matrix in an SSH terminal and change directory to where you placed your workshop source code.
4. Manually compile and run your program to make sure everything works properly:

   ```
   gcc -Wall w6p2.c -o w6 <ENTER>
   ```

   If there are no errors/warnings generated, execute it: ***w6*** *<ENTER>*
5. Run the submission command below (replace **profname.proflastname** with **your professors** Seneca userid and replace **NAA** with your section):

   ```
   ~profName.proflastname/submit 144w6/NAA_p2 <ENTER>
   ```
6. Follow the on-screen submission instructions