For all future communications, please use the following SUBJECT LINE:

`CIS 27 Fall 2023 YourName : Needs/Questions`

Without the above SUBJECT LINE, your emails will be deemed as SPAM/Phishing/Virus and will be deleted.

Turn in:

1. **Exercise #1** – Due on **Thursday, December 14, 2023 by 4:00 pm as Email Submission to class Gmail.**

   Final exam (and homework) formats/conventions/styles are explained in class or provided through Canvas. Students must follow the coding formats, conventions, and styles to obtain full credit for all assignments. More information was available through code demonstration in class — I do coding LIVE during class meetings and discussions, and I use ONLY Visual Studio IDE!

   a) For each exercise, a package must be generated to include the following items:
   - Copy of your source files (C programs → **`*.c, *.h`**)—your source files MUST HAVE A DRIVER NAMED as **`cis27Fall2023FinalExamYourName.c`**
   - Note! Replace "**YourName**" with your own full-name; i.e., **FirstnameLastName**.
   - Copy of output (copy and paste to the end of your program as **`PROGRAM_OUTPUT`** comment block)
   - Copy of **`Logic_Code_Output_Issues`** comment block (as a separate comment block) after the **`PROGRAM_OUTPUT`**.

   b) Emailing each package as follows,
   - One email message for each exercise.
   - The SUBJECT line of the message should have the following line:

     `cis27Fall2023FinalExamYourName.c`

   - Attaching ALL source files (only `*.c` and `*.h`) that was created in Part a) individually. Double check before email.

     `cis27Fall2023FractionYourName.h` `cis27Fall2023FractionYourName.c`

     `cis27Fall2023FinalPolyNodeYourName.h` `cis27Fall2023FinalPolyNodeYourName.c`

     `cis27Fall2023FinalPolyTermYourName.h` `cis27Fall2023FinalPolyTermYourName.c`

     `cis27Fall2023FinalUtilityYourName.h` `cis27Fall2023FinalUtilityYourName.c`

     The above files are also from the setup code!

   - Students may have intentionally emailed something and then claimed emailing by mistake so that they can email again at later time. You will confirm my receiving of your email before leaving so that any of your mistake will have no recourse afterward! No later time email accepted!

   - Do not email ZIP files, image files, *.sln files. You will get ZERO if you email any of these files.

     Note 0! "YourName" means FirstnameLastname—no abbreviation!

2. Reminder!
   - Getting the program to work *is not enough to earn full credit*. Your program must run correctly and follow all proper conventions and consistent styles as explained in class, and produce the exact textual information/display as shown/required (of course, with appropriate user input data/values for each run/selection!) to receive credit accordingly.
   - Also, your program must work with all reasonable data sets or patterns to start with.
   - Again, writing code is not just the code works. It also involves care, patience, coding idioms + forms, and other reminders. Please see the posted code written in class and the coding convention C file.
3. You will get ZERO points if your code does not compile! Please make sure that you compile your code frequently and correctly throughout the working session. Please check and run your submission again exactly as you just submitted.
4. Q.E.D.

****

More Notes!

- You are only allowed to use `printf()` and `scanf()` from the `stdio.h` header.
- You are not allowed to use any other syntax structures that are not introduced in class and class meetings — please confirm with the instructor if you have any doubt!
- You are not allowed to use functions written by someone else.
- You must write all functions yourself before using them in the required class work.
- All local variables must be declared at the top of its function.
- Except for the member data, function arguments and local variables within member functions, and indices of `i`, `j`, `k`, etc., all other variables must have the initials of your first name and last name added to the end of the variable names.

  For examples,

```
int usrInputFL;
int digitCountFL;
int absValueFL;
int tmpFL;
```

- All function names must have the initials of your Firstname and Lastname appended at the end.

  For examples, assuming the name is "`Nice Effort`",

```
void displayClassInfoNE(void);

void runMenuHw2NE(void);
```

- All filenames must have your complete Firstname and Lastname added as required.

- The values of the elements from the created array, it any, MUST NOT be changed/modified at all during the analysis steps and process unless they are required to be updated or changed based on the specified tasks.
- Again, you must declare all local variables at the top of the function; one declaration per statement except for the indices that you may have.
- Please be consistent with the above format requirements.
- Please follow all of the code formats, idioms, and practices that have been presented in class as well as in code and coding convention C file.
- Penalties will be applied for bad coding and practices as discussed/quizzed/demonstrated throughout class.
- After copying the output to the driver, do not manually modify/change/insert text for the output of your code! You will get zero (0) for the work! I will run your code and know if you manually alter the pasted output as submitted in your C file!

Reminder!

- In your program, no GLOBAL DATA are allowed, and you must write all needed functions (no library functions values are allowed).
- Again, writing code is not just the code works. It also involves care, patience, coding idioms + forms, and other reminders. Please see the posted code written in class and the explanation on coding convention/style C file.

Plus More notes!

1. We write code to manipulate data (which are provided by the user) to produce the required outcome in the most efficient way!
2. Getting the program to work is not enough to earn full credit. Your program must run correctly and follow all proper convention and consistent styles as explained in class in order to receive credit accordingly.
3. Writing code is not just the code works. It also involves patience, care and code idioms + forms along with others. Please see the posted code that have been written in class as well as the coding convention C file — posted on Canvas.

And plus more notes!

1. You will get zero (0) points if your code does not compile! Please make sure that you compile your code frequently and properly throughout the working session. Please check and run your submission again exactly as you just submitted.
2. A `runMenuFinalExamFL()` function has a menu that MUST BE a combination of `do-while` and `switch`. Any other form of the menu will receive 0 for the whole homework.
3. You will be penalized heavily if there are violations on code conventions as explained in class, homework submissions, quizzes, and document posted on Canvas!
4. Your code must work with all reasonable data sets/patterns. At least, your code should be tested with the given data samples indicated/given from the exam/document, etc.

5. Pay attention to naming; that means the specified/required names, your own generated names, and filenames. Penalty points will apply if you do not follow the instructions.
6. You are only allowed to use `stdio.h` and `stdlib.h` for `printf()`, `scanf()`, `malloc()`, and `free()`. All other functions must be written by you!
7. All function names must have the initials of your firstname and lastname appended at the end.
8. Q.E.D.

****

Exercise Assignment

Exercise 1 – Due Thursday, December 14, 2023 at 11:45 am by emailing to class EMAIL (to class Gmail)

(1) Write a C program with calls to functions to produce the output given below.

(2) First, the program should display the output to screen as

```
We write code to manipulate data (which are
provided by the user) to produce the
required outcome in the most efficient way!
```

You need to replace "**Your Name**" with your real when you submit your work/email.

The above result should come from a call to a function named as `displayClassInfoFL()`, where `FL` must be replaced by the initial of your first name and the initial of your last name. For examples, if your name is **John Smith** then `FL` should be `JS` throughout all of your work/code as mentioned; the function name `displayClassInfoFL()` will become `displayClassInfoJS()`.

(3) Then, the program will continue with calls to the required functions to display the remaining results. The sample output is given below.

```
// OUTPUT - Sample Run

We write code to manipulate data (which are
provided by the user) to produce the
required outcome in the most efficient way!

For Code Convention and Style -

    Again, writing code is not just the code works.
    It also involves care, patience, coding idioms + forms,
    and other reminders. Please see the posted code written
    in class and the explanation of coding convention
    C/CPP file.

    AND

    If writing in C, you are only allowed to use printf()
    and scanf() from the stdio.h header. You will write
    everything else by yourself.

    If writing in C++, you are only allowed to use
    cout and cin from the iostream header. You will write
    everything else by yourself.

    ALSO,

    - Including PROGRAM COMMENT block!

    - Inserting required Comment lines!
```

- Removing unnecessary blank lines!

- If needed, 1 blank line is sufficient!

- Keeping lines to be reasonable short; for examples, around 65 characters!

- For functions without arguments, insert void in the function prototypes!

- For functions without arguments, remove void in the argument list of the function definitions!

- Keeping indentation levels with the same consistently!

- Declaring all variables at the top of function!

- Declaring one variable per line except for the indices!

- Using better name!

- Following the naming rules!

- Removing all comments that are not required!

- Inserting proper spaces around '(', ')', '{', and operators!

- Using proper code idioms, and efficient operators and styles!

- Paying attention to the exact layout of the required output to earn full credits!

- Replacing FL with the initial of your first-name and last-name!

For Operational Code -

- No global variables/values!

- No extern setups!

- Menu must be a combination of do - while and switch structures, and it has no arguments!

- Use initialization with proper initial value!

- If C++, you must use proper uniform initialization.

- Variables may not need to be initialized!

- If it is C++ coding, then
  "string" class is not allowed in this class. For the

```
    classes that are not written by you, only ostream
    (for cout) and istream(for cin) are allowed.
    You are only allowed to use cout and cin from
    the iostream header.

- Again, uniform initialization!

- All pointers must have values; that means initial
  address or setting address.

- After releasing the dynamic block, its pointer
  must be reset to NULL(in C) or nullptr (in C++).

- Dynamic objects and data->Memory management!

- In C, you are allowed to use only malloc() and
  free().

- In C++, you must use "new" and "delete" operators

- In C/C++, do not write
      someVariable = -1 * somevariable;
      someIncrement = someIncrement + 1;
      if (someTestingValue != 0)
      while (someTestingValue != 0)

- In C/C++, do write
      someVariable = -somevariable;
      someIncrement++;
      if (someTestingValue)
      while (someTestingValue)


- Use proper code idioms, and efficient operators
  and styles!

- Write your own code!

- Except for the member data, member functions,
  function arguments and local variables within
  member functions, and indices of i, j, k, etc., all
  other variables must have the initials of your
  first-name and last name added to the end of
  the variable names.

- For examples,
      int usrInputFL;
      int digitCountFL;
      int absValueFL;
      int tmpFL;

      // Function Prototypes
      void doGood(void);
      void displayDigit(int);
```

```
            // Function Definitions

            void doGood() {

                // Function Body
            }

            void displayDigit(int arg) {

               // Function Body
            }
```

- All stand - alone function names must have the initials of your first-name and last-name appended at the end.

- All filenames must have your complete first-name and last-name appended as required.

For OUTPUT and Copy of OUTPUT -

After copying the output to the driver, do not manually modify / change / insert text for the output of your code! You will get zero (0) for the work! I will run your code and know if you manually alter the pasted output as submitted in your C / C++ (and header) files!

For LOGIC_CODE_OUTPUT_Issues block -

Do not forget this block!

For the final exam, NO ADDTIONAL FUNCTIONS are ALLOWED!

++++++++++

While displayClassInfoFL() is running!

CIS 6 – Introduction to Programming (Using C)
Laney College
Your Name

Information --
  Assignment: Final Exam
  Implemented by: Your Name
  Required Submission Date: 2023/12/14

++++++++++

While displayVersionFL() is running!

Enter your last 2 digits of student ID: 26

26 confirms the following --

```
    I will work with VERSION 0!

++++++++++

***********************************************
*             MENU - Final Exam 0E            *
* 1. Creating 2 Poly's                        *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                     *
***********************************************
Enter an integer for option + ENTER: 6

Wrong Option!

***********************************************
*             MENU - Final Exam 0E            *
* 1. Creating 2 Poly's                        *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                     *
***********************************************
Enter an integer for option + ENTER: 4

Calling printAllPoly() to display ALL Poly's!

  While printAllPoly() is running -

    Empty Poly!

    Empty Poly!

    Empty Poly!

***********************************************
*             MENU - Final Exam 0E            *
* 1. Creating 2 Poly's                        *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                     *
***********************************************
Enter an integer for option + ENTER: 1

Calling runPolySubMenuFL()!

  **********************
  *  Creating Sub-Menu *
  * 1. Creating Poly's *
  * 2. Printing Poly's *
  * 3. Returning       *
  **********************
  Enter an integer for option (1-3): 2

  Printing Poly's!
```

```
For poly1 -

  For poly1, there is/are 0 term(s)!

For poly2 -
  For poly2, there is/are 0 term(s)!

**********************
*  Creating Sub-Menu *
* 1. Creating Poly's *
* 2. Printing Poly's *
* 3. Returning       *
**********************
Enter an integer for option (1-3): 1

Creating NEW Polynomial!

For poly1:

The existing Poly will be REMOVED!
  Enter 1 to continue or anything else to return: 1

removePoly()!

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 3

Enter an int for num: -69196

Enter an int for denom: 515

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 5

Enter an int for num: 52125

Enter an int for denom: 3

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 4

Enter an int for num: -818

Enter an int for denom: 1441
```

```
Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 0

Enter an int for num: 1

Enter an int for denom: -3223

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 2

Returning to previous menu!

Creating NEW Polynomial!

For poly2:

The existing Poly will be REMOVED!
  Enter 1 to continue or anything else to return: 1

removePoly()!

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 2

Enter an int for num: 545

Enter an int for denom: -121

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 1

Enter an int for order: 1

Enter an int for num: 61216

Enter an int for denom: 383

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 2

Returning to previous menu!

**********************
```

```
*  Creating Sub-Menu *
* 1. Creating Poly's *
* 2. Printing Poly's *
* 3. Returning       *
**********************
Enter an integer for option (1-3): 2

Printing Poly's!
For poly1 -

Address: 00C490B8
  termPtr: 00C49080
    termPtr->order: 5
    coeff.num: 17375
    coeff.denom: 1

  next: 00C49D50

Address: 00C49D50
  termPtr: 00C49D18
    termPtr->order: 4
    coeff.num: -818
    coeff.denom: 1441

  next: 00C452E0

Address: 00C452E0
  termPtr: 00C452A8
    termPtr->order: 3
    coeff.num: -69196
    coeff.denom: 515

  next: 00C49DC0

Address: 00C49DC0
  termPtr: 00C49D88
    termPtr->order: 0
    coeff.num: -1
    coeff.denom: 3223

  next: 00000000

For poly1, there is/are 4 term(s)!

For poly2 -

Address: 00C4D1B0
  termPtr: 00C49DF8
    termPtr->order: 2
    coeff.num: -545
    coeff.denom: 121

  next: 00C4D1E8

Address: 00C4D1E8
```

```
    termPtr: 00C4D300
      termPtr->order: 1
      coeff.num: 61216
      coeff.denom: 383

    next: 00000000

  For poly2, there is/are 2 term(s)!

*********************************************
*             MENU - Final Exam 0E          *
* 1. Creating 2 Poly's                      *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                   *
*********************************************
Enter an integer for option + ENTER: 2

Calling displayPalindrome() with 2 Poly's!

  While displayPalindrome() is running -

  Only even order terms will be considered to
  assess and analyze with even digits!

  Poly 1:
    There is/are 2 even order Palindrome term(s)!
    With that, there is/are 3 unique even digits of
      2 seen 2 time(s)
      4 seen 2 time(s)
      8 seen 2 time(s)

  Poly 2:
    There is/are 1 even order Palindrome term(s)!
    With that, there is/are 2 unique even digits of
      2 seen 1 time(s)
      4 seen 1 time(s)

  From both Poly #1 and Poly #2, the uncommon even
  digit(s) is/are
      8

  Digit 8 is seen in the denominator of the term at

  Address: 00C49D50
    termPtr: 00C49D18
      termPtr->order: 4
      coeff.num: -818
      coeff.denom: 1441

    next: 00C452E0

There is/are 0 unique even Palindrome digit(s) seen
in ONLY ONE either numerator or denominator!
```

```
*********************************************
*             MENU - Final Exam 0E          *
* 1. Creating 2 Poly's                      *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                   *
*********************************************
Enter an integer for option + ENTER: 1

Calling runPolySubMenuFL()!

  **********************
  *  Creating Sub-Menu *
  * 1. Creating Poly's *
  * 2. Printing Poly's *
  * 3. Returning       *
  **********************
  Enter an integer for option (1-3): 2

  Creating NEW Polynomial!

  For poly1:

  The existing Poly will be REMOVED!
    Enter 1 to continue or anything else to return: 1

  removePoly()!

  Do you have a term to use?
    Enter 1 to continue or 2 to return: 1

  Enter an int for order: 3

  Enter an int for num: -69196

  Enter an int for denom: 515

  Running addTermToPoly()! --> TODO

  Do you have a term to use?
    Enter 1 to continue or 2 to return: 1

  Enter an int for order: 4

  Enter an int for num: 808

  Enter an int for denom: -1441

  Running addTermToPoly()! --> TODO

  Do you have a term to use?
    Enter 1 to continue or 2 to return: 1

  Enter an int for order: 0
```

```
Enter an int for num: 6

Enter an int for denom: -32023

Running addTermToPoly()! --> TODO

Do you have a term to use?
  Enter 1 to continue or 2 to return: 2

Returning to previous menu!

Creating NEW Polynomial!

For poly2:

The existing Poly will be REMOVED!
  Enter 1 to continue or anything else to return: 2

Returning to previous menu!

**********************
*  Creating Sub-Menu *
* 1. Creating Poly's *
* 2. Printing Poly's *
* 3. Returning       *
**********************
Enter an integer for option (1-3): 2

Printing Poly's!
For poly1 -

Address: 00C4D488
  termPtr: 00C4D370
    termPtr->order: 4
    coeff.num: -808
    coeff.denom: 1441

  next: 00C4D338

Address: 00C4D338
  termPtr: 00C4D220
    termPtr->order: 3
    coeff.num: -69196
    coeff.denom: 515

  next: 00C4D3E0

Address: 00C4D3E0
  termPtr: 00C4D3A8
    termPtr->order: 0
    coeff.num: -6
    coeff.denom: 32023

  next: 00000000
```

```
  For poly1, there is/are 3 term(s)!

  For poly2 -

  Address: 00C4D1B0
    termPtr: 00C49DF8
      termPtr->order: 2
      coeff.num: -545
      coeff.denom: 121

    next: 00C4D1E8

  Address: 00C4D1E8
    termPtr: 00C4D300
      termPtr->order: 1
      coeff.num: 61216
      coeff.denom: 383

    next: 00000000

  For poly2, there is/are 2 term(s)!

    *****
    * Sub-Menu
    * 1. Creating Poly's
    * 2. Printing Poly's
    * 3. Returning
    *****
    Enter an integer for option (1-3):

***********************************************
*             MENU - Final Exam 0E            *
* 1. Creating 2 Poly's                        *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                     *
***********************************************
Enter an integer for option + ENTER: 2

Calling displayPalindrome() with 2 Poly's!

  While displayPalindrome() is running -

  Only even order terms will be considered to
  assess and analyze with even digits!

  Poly 1:
    There is/are 2 even order Palindrome term(s)!
    With that, there is/are 5 unique even digits of
      0 seen 2 time(s)
      2 seen 2 time(s)
      4 seen 2 time(s)
      6 seen 1 time(s)
      8 seen 2 time(s)
```

```
  Poly 2:
    There is/are 1 even order Palindrome term(s)!
    With that, there is/are 2 unique even digits of
      2 seen 1 time(s)
      4 seen 1 time(s)

  From even order term(s) found in both Poly #1 and Poly #2 -
    The uncommon even digit(s) is/are
      0
      6
      8

  Digit 0 is seen in the numerator of the term at

  Address: 00C4D488
    termPtr: 00C4D370
      termPtr->order: 4
      coeff.num: -808
      coeff.denom: 1441

    next: 00C4D338

  Digit 6 is seen in the numerator of the term at

  Address: 00C4D3E0
    termPtr: 00C4D3A8
      termPtr->order: 0
      coeff.num: -6
      coeff.denom: 32023

    next: 00000000

  Digit 8 is seen in the numerator of the term at

  Address: 00C4D488
    termPtr: 00C4D370
      termPtr->order: 4
      coeff.num: -808
      coeff.denom: 1441

    next: 00C4D338

There is/are 1 unique even Palindrome digit(s) seen
in ONLY ONE either numerator or denominator!

*********************************************
*             MENU - Final Exam 0E          *
* 1. Creating 2 Poly's                      *
* 2. Calling displayPalindrome() for 2 Poly's *
* 3. Calling printPoly() to display 2 Poly's  *
* 4. Quit                                   *
*********************************************
Enter an integer for option + ENTER: 4
```

```
Have a great Holiday Season!
```

Your program must have and use ONLY the following functions,

```
void displayClassCodingStatementFL(void);
void displayClassInfoFL(void);
void displayVersionFL(void);
void runMenuFinalFL(void);
void displayDigitDetailFL(int);
```

(4) Save the program as required ; and

(5) The above output should be copied to the PROGRAM_OUTPUT comment block.

In the output from your program –

a. You should be careful about the placement and format (spacing, indentations, blank lines, and characters) of your program output.

b. The above output should be copied and added to the PROGRAM OUTPUT comment block. No manual manipulation to the output is allowed; you will get ZERO for the whole work if any manual/hand manipulation is found!

.

Email the source files (your program) above using the SUBJECT LINE of

Cis27Fall2023FinalExamYourName.c