College of Computer Studies
De La Salle University

COMPRO2 Machine Project
Shopping Kiosk

Release Date: Feb. 22, 2017 (W)

Life in the 21st century is characterized by the need of people for convenience and for fast service. Thus, a lot of online shopping systems abound. This allows people to shop when they want and at the comfort of their own homes or wherever they may be. Internet connection allows them to stay connected and to do the tasks they need to do (even including shopping). This even allows them to compare prices and/or to search for promotions. It should also be noted that the online shopping model can also be applied to facilitate donations from individuals who want to pledge items (or cash equivalents of certain items) for specific beneficiaries (e.g., victims of calamities or resource-poor communities).

Specifications

For this project, we will focus on a simple computerized shopping system where there are basically two main types of users: (1) Administrator and (2) Shopper. You may think of this computerized system as one that may be deployed in a kiosk (similar to an ATM). Here are the features (described per screen / menu): *[**Warning** this is a long list, but do not be overwhelmed. Some of our class discussions will be anchored on the MP, thus some of the features will be used as examples in our discussion. Therefore, ideally, if you constantly work on the project (incrementally, rather than cramming everything at the end), you will be able to fix any problems that may arise and would hopefully allow you to present a better resulting final MP.]*

Note that there are unlimited number of users who may sign up for accounts. There may also be unlimited number of stock information. These information should be stored as linked structures.

Main Menu

- ***Login***
  The username and password is asked from the user. If the username and password matches an entry in the Users List, the user proceeds to the next menu screen. If the user is an administrator, the next screen is the Administrator Menu. If the user is a registered shopper, the next screen is the Shopper Menu. If the username is found in the Users List, but the password does not match, the system should give an appropriate message and to allow the user to try logging in again. On the third attempt at logging in to the system using an incorrect password (for the same user), the user account is locked (i.e., cannot access account even if he logs in using the correct password in a fourth [fifth, …] trial) and a message like "Contact Administrator" or "Call Hotline to unlock account" is shown.
- ***Sign Up***
  The user is asked to input the following information to be stored into the Users List:
  - Username – should be unique (that is, it should not exist yet as a username in the Users List) and should contain at least 3 characters and at most 15 characters. Non-compliance with the required input should result to the program asking for another username again until the user complied. Note that this is done immediately (i.e., password and other information is not yet asked). Assume the username is one word only (i.e., no spaces).
  - Password – should contain at least 6 characters and at most 15 characters, at least one of which is not a letter. Non-compliance with the requirement should result to the program asking for another password until the user complied. Checking for compliance and asking for new inputs for those non-compliant is done when all the user info is given already. Note that only the non-compliant should be asked again. Assume that the password is one word only.
  - User Info – consists of the
    - Name – further consists of first name, middle name, and last name. Each of which is a string that can accommodate at most 20 characters as input.
    - Address – string containing at most 50 characters as input.

  Once all the inputs are acceptable (based on requirements stated), the user is asked to input type of account to be created. For example: "Create [S]hopper or [A]dministrator account?". Answer to this question can be a capital or small letter. If 's' or 'S', then a shopper account is created. Other information for a shopper are also initialized (i.e., Credit Limit is initialized to PhP5000.00, Outstanding Balance is initialized to PhP0.00, Cart is

initially empty but would later contain at most 100 item types. To avoid duplication of entries, each item type will consist of just the project code and the quantity and any other information we need for later display will just be referenced through the series of stock information). If 'a' or 'A', the user is prompted to enter authorization code and when authorization code is entered successfully, the user is noted as an administrator. For simplicity, authorization code is "DLSU2017" and this is case sensitive. After account creation, the user is directed back to the Main Menu (log-in / sign-up). Note that there can be unlimited number of users (both administrator and shopper), thus implementation for the list of users will be via linked list.

Administrator Menu

- ***Manage Accounts Menu***
  - **View Locked Accounts**
    This menu option displays the usernames and whole names of accounts that have been locked. The display is arranged in alphabetical order based on the username.
  - **Unlock Specific Account**
    This menu option first displays the usernames and whole names of accounts that have been locked, in alphabetical order based on username. Then, the program prompts the administrator to input the username of the account to unlock. Note that once the account is unlocked, the counter for trials for incorrect password for that account is reset back to 0 and would, in effect, allow the user to log-in using his username and password again.
  - **Unlock All Locked Accounts**
    This menu option first displays the usernames and whole names of accounts that have been locked, in alphabetical order based on username. Then, the program automatically resets the counter for trials for incorrect password for these accounts back to 0 and would, in effect, allow the users to log-in using their respective usernames and passwords again.
  - **View Accounts with Outstanding Balance**
    This menu option displays the usernames, whole names of shoppers, and their outstanding balances in alphabetical order based on username. Note that those shoppers whose outstanding balance is 0.00 should not be displayed.
  - **Return to Administrator Menu**
    This menu option first displays the usernames and whole names of accounts that have been
- ***Manage Stocks Menu[S]***
  - **Add New Stock**
    This menu option asks for 1 stock information as input, then goes back to the Manage Stocks Menu. Each stock information is organized by category and there are at most 10 categories, thus each entry of the stock information contains the following:
    - Category – one word string input that is at least three characters and at most 15 characters.
    - Product List – this is an unlimited list of products under the given category, each of which consists of the following:
      - Supplier – a phrase input that is at most 15 characters representing the source of the product.
      - Product – a phrase input that is at most 15 characters, this represents the product. This should be a unique entry under the given category and supplier. Meaning, we can have: (1) Category: Food, Supplier: ABC, Product: Soy Sauce; (2) Category: Food, Supplier: ABC, Product: Soy Sauce-Light; (3) Category: Food, Supplier: XYZ, Product: Soy Sauce-Light; (4) Category: PromoFood, Supplier: ABC, Product: Soy Sauce–Light. But, if input is again (for example): Category: Food, Supplier: ABC, Product: Soy Sauce-Light, then a message will be displayed on the screen indicating that this already exists and instruct the administrator to use the Modify option, Restock, or to Add New Stock again but give a different product name. In which case, after the message, the administrator is brought back to the Manage Stocks Menu (note that the quantity available, etc. are not asked as input anymore and no duplicate entry is created).
      - Quantity Available – integer input representing the count of this product that is currently available
      - Purchase Price – floating point input representing price of each of this product as given by supplier

- Unit Selling Price – floating point input representing the normal selling price of this product (before any discount, if applicable)
- Discount Rate – floating point input representing a percentage of discount from the selling price to be given to the shopper. Example, if discount rate is 12.5, this is 12.5% discount.
- Quantity Sold – integer value initially set to 0.
- Product Code – a unique 8-character string generated to easily refer to the product (i.e., instead of asking for a long string for the product later, this is the input that will be expected). It is automatically generated by using the first letter of the category, followed by the first character in the Supplier, the first character in the Product, and a randomly generated five numeric characters. For example: FAS08108

View All Stocks

This menu option displays all current information about the stocks. The display should be in table format and are arranged by category, then by supplier, then by product code. Make sure the values are aligned (use the maximum number of characters as basis in the allotment of space, assume quantities are at most 4 digits, prices have 2 decimal places, and discount rate is always 1 decimal place). Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Supplier	Product Code	Product	Qty Available	Qty Sold	Purchase Price	Selling Price	Discount Rate

View Stocks by Category

This menu option asks the category as input. If the category does not exist, then a message is displayed. If it exists, this feature displays all current information about all the stocks in the given category. The display should be in table format and are arranged by supplier, then by product code. Make sure the values are aligned. Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Supplier	Product Code	Product	Qty Available	Qty Sold	Purchase Price	Selling Price	Discount Rate

View Stocks to Reorder

This menu option displays all current information about all the stocks that have to be replenished (i.e., quantity available is at most 5). This will give information on what to reorder. The display should be in table format and are arranged by category, then by supplier, then by product code. Make sure the values are aligned. Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Supplier	Product Code	Product	Qty Available	Qty Sold	Purchase Price	Selling Price	Discount Rate

Modify Stock Info

This menu option first displays all current information about all stocks similar to how View All Stocks does the display, then asks the administrator for the product code of the item that will be modified. If product code is not found, a message is displayed and the administrator is brought back to the Manage Stocks Menu. If product code is found, the administrator is prompted to identify which of the following will be modified until he chooses to end modification of the current product. Meaning:

[1] Modify Category

- The effect is the product will be moved to be under this new category. If the category does not exist yet, a new category is created too.

[2] Modify Supplier

- The effect is the product will have a new supplier.

[3] Modify Product

- The effect is the product will have a new product name.

[4] Modify Purchase Price

- The effect is the purchase price will be updated based on the new input. All succeeding computations (or generations of report will use the current values).

[5] Modify Selling Price

- The effect is the selling price will be updated. All succeeding computations (or generations of report will use the current values).

[6] Modify Discount

- The effect is the discount rate will be updated. All succeeding computations (or generations of report will use the current values).

[7] Finish Modification of Product <Product Code>

- Once this option is chosen, it is possible that a new product code is generated (that is if the category, supplier, and/or product is changed). Meaning if the new category does not have the same first letter then the product code's first letter should be updated to reflect the new first letter of the category. Also, if the new supplier has a different first letter then there is a need to change the second letter in the product code. And if the new product does not have the same first letter, the third letter of the product code should be modified, as well. If the updated product code is no longer unique (due to the modifications), then regenerate the last five digit numeric characters. Make sure to inform the user of the change in the product code, if applicable. The administrator is brought back to the Manage Stocks Menu.

- **Restock**

This menu option first displays all current information about all stocks similar to how View All Stocks does the display, then continuously asks the administrator for the product code and additional quantity for that item until the administrator inputs a product code that is not found (e.g., XXX). Do not forget to end with a displayed message "Last input product Code not found: Not restocked", then bring the administrator back to the Manage Stocks Menu.

- **Save Inventory**

This menu option asks the administrator for the file name where the stock information will be saved to. If the file exists, this option will overwrite whatever is in the existing file. In saving, the entries are arranged in alphabetical order of category, then by product code. This save option saves all current information as a text file strictly following the format below:

```
<category A><space><product code A1><space><product A1><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
<category A><space><product code A2><space><product A2><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
: : :
<category A><space><product code An><space><product An><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
<category B><space><product code B1><space><product B1><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
<category B><space><product code B2><space><product B2><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
: : :
<category B><space><product code Bn><space><product Bn><new line>
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
: : :
<category Z><space><product code Zn><space><product Zn><new line>
```

```
<quantity available><space><purchase price><space><supplier><new line>
<selling price>&<discount rate><space><quantity sold><new line>
<new line>
<end of file>
```

- **Update Inventory from File**
  This menu option first asks the administrator the file name of the source information. This option assumes that the file is a text file and is in the exact same format as indicated in the Save Inventory option. Note that if the entry does not exist in the currently running program, then a new entry is created. If an entry exists (i.e., the product code already exists), the current data in the running program is shown, together with the one from the one extracted from the file, then the administrator is asked which one will be retained. If the administrator chose 'P', then retain the one in the currently running program, but if the administrator chose 'F', the data for that "duplicate" entry in the currently running program is removed and the new data from the file is used.

- ***Prepare Delivery Receipt***

This menu option generates the delivery receipt for each Shopper with purchases. Assume that after the delivery receipt is generated, products are delivered.

On the screen, the following information per shopper is shown. Arrangement of the items are in alphabetical order based on product code. Allow the administrator to press 'N' to view the next shopper's delivery receipt, press 'B' to view the previous shopper's delivery receipt, and 'X' to exit the viewing.

User ID : <Username>
Customer Name : <Last Name>, <First Name> <Middle Name>
Delivery Address: <Address>

Product Code	Product	Quantity	Unit Price	Total Price		Item Subtotal
<product code 1>	<product 1>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>
<product code 2>	<product 2>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>
:::						
<product code n>	<product n>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>

Number of items : <sum of all quantity>
Total Discount : PhP <total amount of discount>
Bill Amount : PhP <total after discount>
Total Outstanding: PhP <total after discount + previous outstanding balance>

The delivery receipts are saved to a text file too. The file name is asked from the administrator. If the file exists, the contents will be overwritten. The format for saving is strictly as follows:

```
<user id 1><space><number of ordered products><new line>
<qty 1><space><product code 1><space><product 1><new line>
@<unit price 1>&<discount rate 1><new line>
<qty 2><space><product code 2><space><product 2><new line>
@<unit price 2>&<discount rate 2><new line>
```

```
<qty 3><space><product code 3><space><product 3><new line>
@<unit price 3>&<discount rate 3><new line>
: : :
<qty n><space><product code n><space><product n><new line>
@<unit price n>&<discount rate n><new line>
<new line>
<user id 2><space><number of ordered products><new line>
<qty 1><space><product code 1><space><product 1><new line>
@<unit price 1>&<discount rate 1><new line>
<qty 4><space><product code 4><space><product 4><new line>
@<unit price 4>&<discount rate 4><new line>
: : :
<qty m><space><product code m><space><product m><new line>
@<unit price m>&<discount rate m><new line>
<new line>
: : :
<user id x><space><number of ordered products><new line>
<qty 4><space><product code 4><space><product 4><new line>
@<unit price 4>&<discount rate 4><new line>
: : :
<qty y><space><product code y><space><product y><new line>
@<unit price y>&<discount rate y><new line>
<new line>
<end of file>
```

As the assumption is that the delivery is made after the generation of the receipt, the program now resets all orders of the shoppers after this option [whether viewed or not] (i.e., list of products are now empty) and the bill amount is now added to the outstanding balance of the shopper.

- ***Shutdown Kiosk***
  This option terminates the program properly.
- ***Log Out***
  This will just allow the administrator to sign out. The Main Menu is then shown (for the next user to log-in or sign up).

Shopper Menu [Many of the features here will just need to call the same function as those indicated already in the Administrator Menu; tweaking the said functions may be necessary to accommodate the slight differences.]
Upon entry to this screen, the shopper is welcomed. He/she is shown his/her outstanding balance. Then, the menu options below are shown and the shopper is allowed to go through each option/suboption repeatedly until he/she chooses to log out.

- ***Modify User Info***
  - **Change name**
    This option will allow the shopper to update his/her first name, middle name, and last name.
  - **Change address**
    This option will allow the shopper to update his/her address.
  - **Change password**
    This option will allow the shopper to change his/her password. Note that rules for a proper password should still be followed.
- ***Browse All Products***
  This menu option displays all current information about all available products (i.e., quantity available should at least be 1). The display should be in table format and are arranged by category, then by supplier, then by product code. Make sure the values are aligned (use the maximum number of characters as basis in the allotment of space, assume quantities are at most 4 digits, prices have 2 decimal places, and discount rate is always 1 decimal place). Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Brand	Product	Product	Qty	Qty	Unit	Discount

	(this is actually the supplier)	Code		Available	Sold	Price (this is actually the selling price)	Rate

Note that this is similar to the View All Stocks option, but instead of "Supplier", "Brand" is indicated and instead of "Selling Price", "Unit Price" is indicated. Also, purchase price (from supplier) is not displayed.

- ***Browse Products by Category***

  This menu option asks for the category he/she wants to browse through and displays all current information about all available products (i.e., quantity available should at least be 1). The display should be in table format and are arranged by supplier, then by product code. Make sure the values are aligned. Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Brand	Product Code	Product	Qty Available	Qty Sold	Unit Price	Discount Rate

  Note that this is similar to the View Stocks by Category option, but instead of "Supplier", "Brand" is indicated and instead of "Selling Price", "Unit Price" is indicated. Also, purchase price (from supplier) is not displayed.

- ***Browse Products on Sale***

  This menu option displays all current information about all available products (i.e., quantity available should at least be 1) and there is an offered discount (i.e., discount rate is more than 0.0%). The display should be in table format and are arranged by category, then by supplier, then by product code. Make sure the values are aligned Make sure to display at most 20 entries only per "page", so that the data would not just flash. Allow the user to press ENTER key to go to the next page. First line in each "page" should be as follows:

Category	Brand	Product Code	Product	Qty Available	Qty Sold	Unit Price	Discount Rate

- ***Add to Cart***

  This option is only shown (and is only available) if the shopper's outstanding balance is less than the credit limit and his/her cart is not yet locked until delivery (see Check Out). In this option, the user is prompted to input the product code and the quantity that he/she wants to purchase. Checking should be done to ensure that the product code is existing and that the quantity is available for purchase. If these are not valid, the user is not allowed to add it to his/her cart. Whether or not an item was added to the cart, the user is asked if he/she wants to add another item to the his/her cart. If [y]es, then he/she is prompted for the product code and the quantity again, and so on. If eventually, the user said [n]o, then the shopper is brought back to the Shopper Menu.

- ***View Cart***

  This option is only shown (and is only available) if the shopper has chosen to add at least 1 product to his/her cart. In this option, those items that the user was able to successfully add to his/her cart will be displayed on the screen in table format with the following first line:

Product Code	Product	Quantity	Unit Price	Total Price		Item Subtotal
<product code 1>	<product 1>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>
<product code 2>	<product 2>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>
: : :						
<product code n>	<product n>	<order qty>	<current selling price>	<order qty * current selling price>	[-<discount rate>%]	<(selling price – discount rate / 100) * qty>

Number of items : <sum of all quantity>
Total Discount : PhP <total amount of discount>
Cart Amount : PhP <total after discount>

Note that the display here is similar to parts of those in Prepare Delivery Receipt, except that instead of "Bill Amount", "Cart Amount" is displayed. Also, some of the other information is not displayed anymore.

After displaying the above information, the following options are displayed:

- **Check out[S]**
  This option means that the user is confirming to purchase everything in his/her cart. Thus, the items chosen are considered sold to this shopper. The quantities sold and quantities available for each of the items chosen are then updated in the inventory/stocks. After this option, the user is not allowed to Add to Cart anymore, until his/her cart contents are cleared (i.e., when items are "delivered"). The user is brought back to the Shopper Menu (but again, Add to Cart is not available). Note that prior to choosing Check out option, the user can still Add to Cart, View Cart, Add to Cart, Edit Cart, etc.
- **Edit Cart Items[S]**
  This option allows the user to modify what is currently in the shopper's cart until the shopper chooses to go back to the View Cart Menu. Meaning:
  [1] Remove item
  - The product code is asked as input and the corresponding product is removed from the shopping cart.
  [2] Update quantity
  - The product code is asked as input and the corresponding new quantity is asked. Note that the new quantity should still be within the limits of the quantity available for that product. If the new quantity is 0, the product is removed from the shopping cart.
  [3] Back to View Cart Menu
  - The effect of this menu option is it terminates the editing of the cart items and brings the shopper back to the View Cart Menu.
- **Back to Shopper Menu**
  The effect of this option is to terminate the View Cart Menu and bring the shopper back to the Shopper Menu.

- ***Settle Outstanding Balance***
  This option is not shown (and is not available) if the shopper's outstanding balance is PhP0.00. In this option the shopper is shown his outstanding balance and is prompted to input his/her credit card information and amount to settle (may be at most equal to the outstanding balance [but can be less]).

After which, the program assumes that payment is successful and the outstanding balance is updated. The shopper is brought back to the Shopper Menu.

- ***Log Out***
  This will just allow the shopper to sign out. For simplicity, for this project, if the shopper fails to Check Out his/her cart before logging out, everything in his/her cart will be lost (i.e., they should be reset). The Main Menu is then shown (for the next user to log-in or sign up).

Bonus

A **maximum** of **10 points** may be given for features **over & above** the requirements (such as use of modules in programming or other features not conflicting with the given requirements or changing the requirements [example: (1) showing sales report on statistics on top selling items, total sales, and total profit; (2) using top selling items and items with discounts to generate recommended items [top picks] to shoppers; (3) saving transactions from multiple days and comparing the sales of each day] subject to **evaluation** of the teacher. **Required features** must be **completed first** before bonus features are credited. Note that use of conio.h, or other advanced C commands/statements may **not** necessarily merit bonuses.

Submission & Demo

- **Phase 1 Deadline: March 8, 2017 (W), first 15 minutes of class time**. Should submission be done after the first 15 minutes of class time of March 8, 2017, but until 5 pm of the said day, an automatic deduction of 30 points will be applied to the Phase 1 grade. However, after the March 8, 5pm (with deduction), the project will not be accepted anymore and the phase 1 grade is automatically 0

  Requirements:
  - Implementation of the following features:
    - All features of Manage Accounts Menu
    - All features of Main Menu
    - Add New Stock feature is limited to one Product per Category for Phase 1
    - View All Stocks and Browse All Products
    - Modify Stock Info
    - Restock
    - Shutdown Kiosk
    - Log out
    - Modify User Info
  - Make sure that your implementation has considerable and proper use of arrays/dynamic lists, linked structures, and user-defined functions, as appropriate, if it is not strictly indicated.

- **Phase 2 Deadline: March 27, 2017 (M), first 15 minutes of class time**. Should submission be done after the first 15 minutes of class time of March 27, 2017, but until 5 pm of the said day, an automatic deduction of 30 points will be applied to the phase 2 grade. However, after the March 27, 5pm (with deduction), the project will not be accepted anymore and the phase 2 grade is automatically 0.

  Requirements: Complete Program
  - Make sure that your implementation has considerable and proper use of arrays/dynamic lists, linked structures, files, and user-defined functions, as appropriate, if it is not strictly indicated.

Important Notes:

1. Use **gcc -pedantic** to compile your C program. Make sure you **test** your program completely (compiling & running) in **G302 and G306 labs**.
2. Do not use brute force. Use **appropriate conditional** statements **properly**. Use, **wherever appropriate**, **appropriate loops** & **functions properly**.
3. You **may** use topics outside the scope of COMPRO2 but this will be **self-study**. Goto *label*, exit, break (except in switch), global variables are **not allowed**.
4. Include **internal documentation** (comments) in your program.

5. The following is a checklist of the deliverables:

Checklist:

- ❒ CD (placed in a case) or flash drive [either of which should be properly labeled with name and section], containing at least 5 files:
  - ❒ source code*
  - ❒ function specifications**
  - ❒ test script***
  - ❒ sample stock information text file
- ❒ Print-out of the source code*
- ❒ Print-out of the function specifications** (short bond papers, stapled or in folder)
- ❒ Print-out of the test script***
- ❒ Short brown envelope with one copy of the proof of submission**** attached at the **upper right** hand corner of the back of the envelope, containing all the above requirements.
- ❒ Another copy of the proof of submission**** to be signed and returned to you
- ❒ email the source code* and function specifications** as attachments to YOUR own email address on or before the deadline

Legend:

*****Source Code also includes the internal documentation. The first page of the source code printout should have the following declaration (in comment), the **printout of which should be signed**:

```
/*************************************************************************************************************************
This is to certify that this project is my own work, based on my personal efforts in studying and applying the concepts
learned. I have constructed the functions and their respective algorithms and corresponding code by myself. The program
was run, tested, and debugged by my own efforts. I further certify that I have not copied in part or whole or otherwise
plagiarized the work of other students and/or persons.

                                                        <your full name>, DLSU ID# <number>
*************************************************************************************************************************/
```

******Function Specifications should be in a table format.

Sample is shown below.

Function Name	Description	Input Parameter	Return Data
playGame	allows the user to play the game with nRow and nCol dimension of the board	nRow – initialized row dimensions of the board nCol – initialized col dimensions	*pWin – updated value of number of wins of player

*******Test Script should be in a table format. There should be 3 categories (as indicated in the description) of test cases **per function**. There is no need to test functions which are only for screen design (i.e., no computations/processing; just printf).

Sample is shown below.

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
sortIncreasing	1	Integers in array are in increasing order already	aData contains: 1 3 7 8 10 15 32 33 37 53	aData contains: 1 3 7 8 10 15 32 33 37 53	aData contains: 1 3 7 8 10 15 32 33 37 53	P
	2	Integers in array are in decreasing order	aData contains: 53 37 33 32 15 10 8 7 3 1	aData contains: 1 3 7 8 10 15 32 33 37 53	aData contains: 1 3 7 8 10 15 32 33 37 53	P
	3	Integers in array are combination of positive and negative numbers and in no particular sequence	aData contains: 57 30 -4 6 -5 -33 -96 0 82 -1	aData contains: -96 -33 -5 -4 -1 0 6 30 57 82	aData contains: -96 -33 -5 -4 -1 0 6 30 57 82	P

********Proof of submission should have the following data:

Name	:	
Section	:	Sxxx
Submitted To	:	Ms. Nathalie Rose Lim-Cheng
Received by	:	
Date/Time Rcvd.	:	

6. Upload the softcopies via Submit Assignment in Canvas. Send also to your **mylasalle account** a **copy** of your source code & function specification by the **deadline**.
7. Use **<surnameFirstInit>.c** & **< surnameFirstInit >.doc** as your filenames.
8. During the MP **demo**, the student is expected to appear on time, to answer questions, in relation with the output and to the implementation (source code), and/or to revise the program based on a given demo problem. Failure to meet these requirements could result to a grade of 0 for the project.
9. It should be noted that during the MP demo, it is expected that the program can be compiled successfully and will run. If the program does not run, the grade for the project is automatically 0. However, a running program with complete features may not necessarily get full credit, as implementation (i.e., code) and other submissions (e.g., function specs and internal documentation) will still be checked.
10. This is an **individual** project. Any form of **cheating** (**working in collaboration**, **asking other people's help**, **copying any part of other's work**, etc.) can be punishable by a grade of **0.0** for the **course** & a **discipline case**. Thus, do not risk it; the consequences are not worth the risk and not worth the nagging reminder by your conscience.

Any requirement not fully implemented or instruction not followed will merit deductions.