**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 1: Programming | | |
| **Submission date** | 20/08/2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Cao Lương Bảo Lâm | **Student ID** | GCD210240 |
| **Class** | Summer2022.Part2.1618 - GCD1101 | **Assessor name** | Lý Quỳnh Trân |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Lâm |

**Grading grid**

|  |  |  |
| --- | --- | --- |
| P1 | M1 | D1 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Lecturer Signature:** | | |

Table of Contents

[Table Of Figure 3](#_Toc111962566)

[**CHAPTER I: PROBLEM STATEMENT (P1)** 4](#_Toc111962567)

[1. Scenario 4](#_Toc111962568)

[2. Tasks 4](#_Toc111962569)

[**CHAPTER II: ALGORITHM (P1 – M1)** 4](#_Toc111962570)

[1. Simple business problem 4](#_Toc111962571)

[2. Definition of algorithm 4](#_Toc111962572)

[a. Definition 4](#_Toc111962573)

[b. Example of algorithm 5](#_Toc111962574)

[3. Properties of algorithms 6](#_Toc111962575)

[4. The algorithm for the problem above 6](#_Toc111962576)

[5. Steps in building the application 7](#_Toc111962577)

[6. The final product 8](#_Toc111962578)

[7. The code 11](#_Toc111962579)

[**CHAPTER III: STEPS TO WRITING A PROGRAM (M1)** 13](#_Toc111962580)

[1. Understand the problem you are trying to solve 13](#_Toc111962581)

[2. Draw a flow chart 13](#_Toc111962582)

[3. Write pseudo-code 14](#_Toc111962583)

[4. Replace pseudo-code with real code 15](#_Toc111962584)

[5. Test and debug 17](#_Toc111962585)

[**CHAPTER IV: EVALUATION (D1)** 22](#_Toc111962586)

[**REFERENCES** 23](#_Toc111962587)

# Table Of Figure

[Figure 1: Flowchart factorial 5](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962813)

[Figure 2: Program Development Life Cycle 7](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962814)

[Figure 3: Console menu-base 8](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962815)

[Figure 4:Code menu-base 8](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962816)

[Figure 5: ADD A BOOK 8](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962817)

[Figure 6: DISPLAY BOOKS 9](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962818)

[Figure 7: SEARCH BY ID 9](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962819)

[Figure 8: SEARCH BY TITLE 9](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962820)

[Figure 9: SEARCH BY AUTHOR 10](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962821)

[Figure 10: UPDATE A BOOK 10](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962822)

[Figure 11: REMOVE A BOOK 10](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962823)

[Figure 12: EXIT PROGRAM 10](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962824)

[Figure 13: MENU-BASE FLOWCHART 13](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962825)

[Figure 14: TEST-CASE 1 18](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962826)

[Figure 15: TEST-CASE 2 19](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962827)

[Figure 16: TEST-CASE 3 19](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962828)

[Figure 17: TEST-CASE 4 20](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962829)

[Figure 18: TEST-CASE 5 20](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962830)

[Figure 19: TEST-CASE 6 21](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962831)

[Figure 20: TEST-CASE 7 21](https://vngreenwich-my.sharepoint.com/personal/lamclbgcd210240_greenwich_edu_vn/Documents/Documents/1618/1_Unit%201%20-%20Assignment%201%20frontsheet%20-%20GCD210240%20-%20Cao%20Lương%20Bảo%20Lâm.docx#_Toc111962832)

## **CHAPTER I: PROBLEM STATEMENT (P1)**

# **Scenario**

You have applied for a post as a trainee with a software development company and have been invited for an interview. You have been asked to demonstrate your problem solving and basic programming skills. To do this you have to prepare a report on using algorithms to solve problems.

You need to explain, using examples, how algorithms are used to solve simple business problems and the steps needed to be followed to produce a working program solution. You should make clear your assumption about your program. The problems to be solved will involve basic procedural programming instructions - sequence instructions (input, output and assignment statements), loops, conditional statements. Problems should be analyzed and designed by the use of flowchart and demonstrated by the use of modules (procedures).

# **Tasks**

• State your simple business problems to be solved.

• Analyze the problem and design the solutions by the use of suitable methods.

• Demonstrate the compilation and running of a program

• Evaluate how the problem is solved from the designed algorithm to the execution program

written by a specific programming language.

## **CHAPTER II: ALGORITHM (P1 – M1)**

# Simple business problem

The libraries in today’s schools are mostly recorded and sorted by hand, and the performance is abysmal. Therefore, having a Library Management System application to handle the primary housekeeping functions of a library is necessary. When these algorithms are applied, they will lead to the final results as expected.

# Definition of algorithm

# Definition

An algorithm is a procedure used for solving a problem or performing a computation. Algorithms act as an exact list of instructions that conduct specified actions step by step in either hardware- or software-based routines.

# Example of algorithm

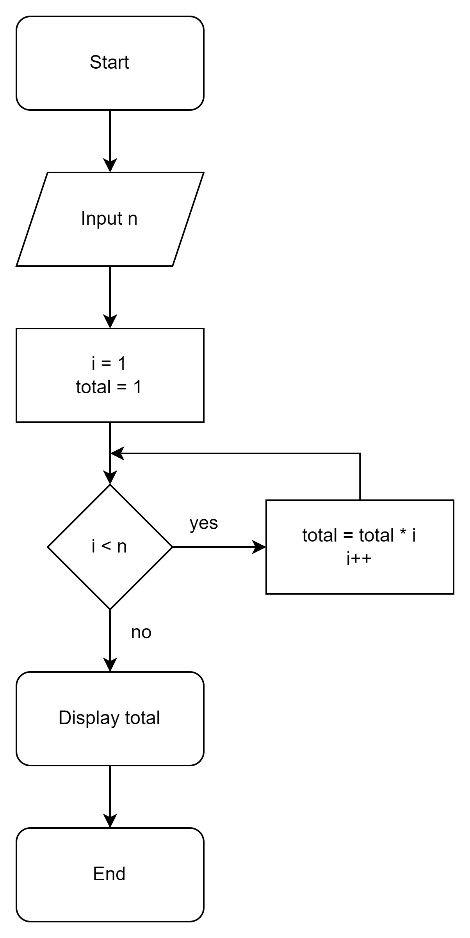
* Problem: Write an algorithm to solve the factorial: n! = 1 \* 2 \* 3 \* 4 \*… \* n (n is integer)
  + Input: n
  + Process: Use algorithm to solve n!
  + Output: the factorial of n
* The algorithm can be represented in two ways, list step and flowchart:
  + List step:
    - Step 1: Input n, total = 1
    - Step 2: i = 1
    - Step 3: total = total \* i, i = i+1
    - Step 4: Go to step 3 if i < n, else continues to step 5
    - Step 5: Display total
    - Step 6: End
  + Flowchart:

Figure 1: Flowchart factorial

# Properties of algorithms

1. Input:

* An algorithm may have many inputs or no inputs at all.

1. Output:

* It should result at least one output.

1. Finiteness:

* An algorithm should have finite number of steps and it should end after a finite time.

1. Effectiveness:

* Each step must be simple and should take a finite amount of time.

1. Definiteness:

* Each step must be clear, well-defined and precise. There should be no any ambiguity.

# The algorithm for the problem above

* Start
* Declare the list libraryManager variable
* Menu algorithm:
  + Display menu options
  + Read option
  + While (option != 8)
  + Switch case (option)
    - Case 1: Add book
    - Case 2: Display book
    - Case 3:
      * Declare integer ID then input ID
      * Search ID then display the book that contains ID
      * End
    - Case 4:
      * Declare string Title then input Title
      * Search Title then display the book that contains Title
      * End
    - Case 5:
      * Declare string Author then input Author
      * Search Author then display the book that contains Author
      * End
    - Case 6:
      * Input ID (No need to declare ID because of case 3)
      * Update the book that contains the ID
      * End
    - Case 7:
      * Input ID (No need to declare ID because of case 3)
      * Remove the book that contains the ID
      * End
    - Default:
      * Display asking the user to reinput option
      * End

# Steps in building the application

There are 6 steps in developing a software or updating the current software:

Step 1: Problem Definition

Step 2: Problem Analysis

Step 3: Algorithm Development

Step 4: Coding & Documentation

Step 5: Testing & Debugging

Step 6: Maintenance

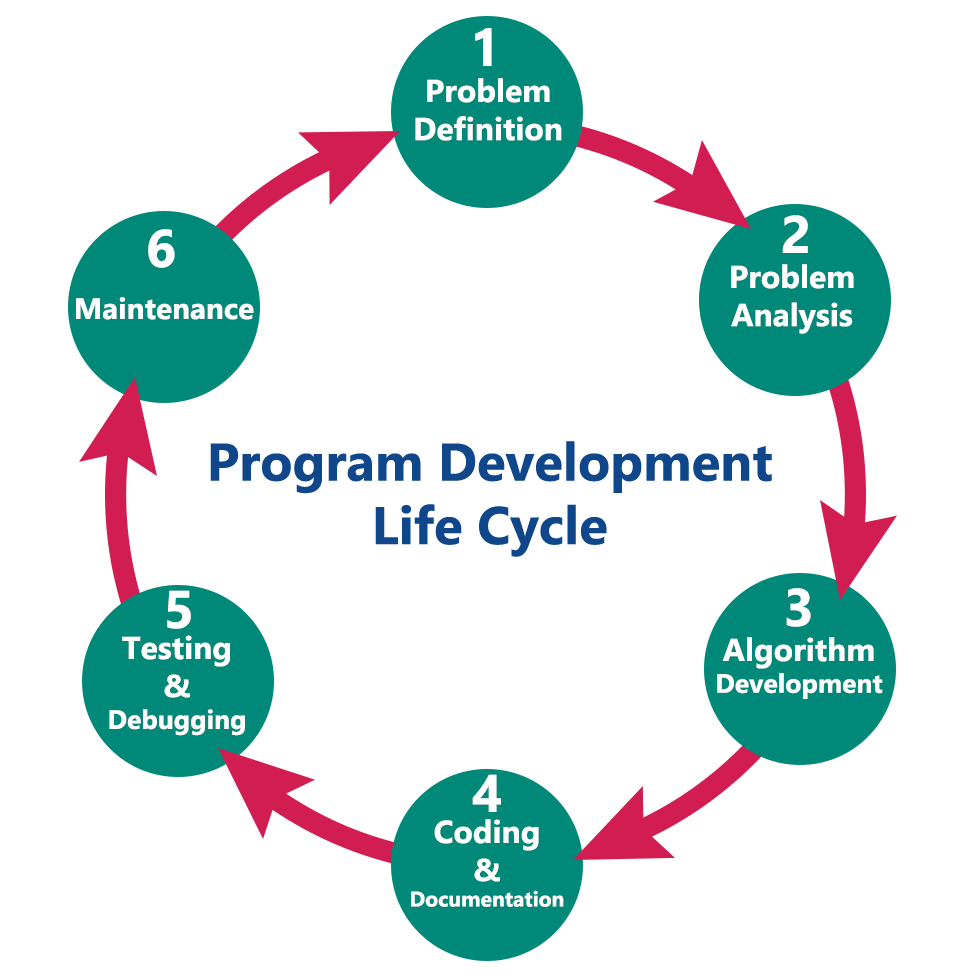


Figure 2: Program Development Life Cycle

# The final product

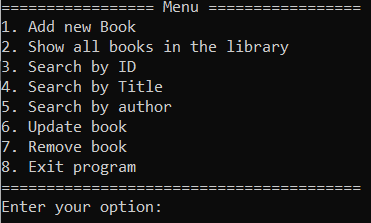
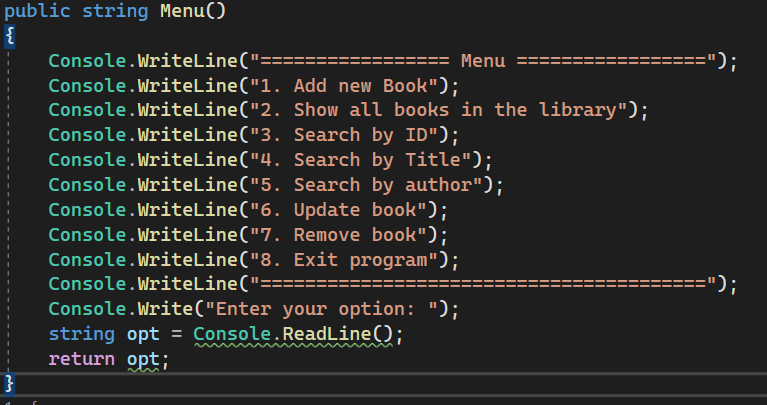
* Print menu-base:

Figure 3: Console menu-base

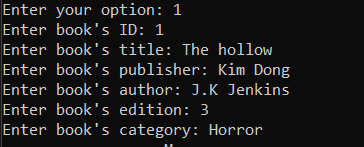
* Option 1: Add a book

Figure 4:Code menu-base

Figure 5: ADD A BOOK

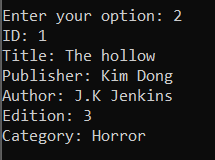
* Option 2: Display books (There is currently only 1 book, but you can add more books using option 1)

Figure 6: Display books

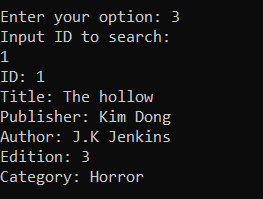
* Option 3: Search by ID

Figure 7: SEARCH BY ID

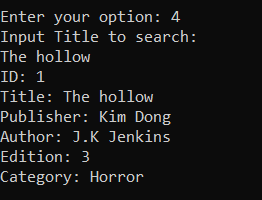
* Option 4: Search by title

Figure 8: SEARCH BY TITLE

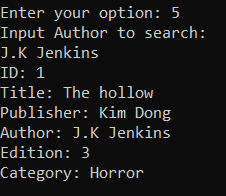
* Option 5: Search by author

Figure 9: SEARCH BY AUTHOR

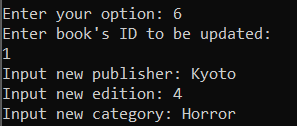
* Option 6: Update a book

Figure 10: UPDATE A BOOK

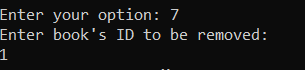
* Option 7: Remove a book

Figure 11: REMOVE A BOOK

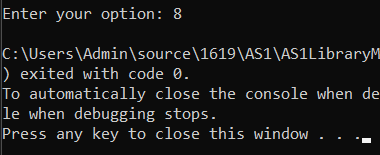
* Option 8: Exit program

Figure 12: EXIT PROGRAM

# The code

* Program.cs:

|  |
| --- |
| namespace AS1LibraryManagementSystem  {  internal class Program  {  static void Main(string[] args)  {  LibraryManager libraryManager = new LibraryManager();  string otp = "0";  do  {  otp = libraryManager.Menu();  switch (otp)  {  case "1":  libraryManager.addBook();  break;  case "2":  libraryManager.displayBooks();  break;  case "3":  Console.WriteLine("Input ID to search: ");  int ID = int.Parse(Console.ReadLine());  Library id = libraryManager.IDSearch(ID);  if (id != null)  {  id.printInfo();  }  else  {  Console.WriteLine("Cannot find ID!");  }  break;  case "4":  Console.WriteLine("Input Title to search: ");  string str = Console.ReadLine();  List<Library> list = libraryManager.TitleSearch(str);  if (list.Count > 0)  {  foreach (var item in list)  {  item.printInfo();  }  }  else  {  Console.WriteLine("Cannot find Title!");  }  break;  case "5":  Console.WriteLine("Input Author to search: ");  string str2 = Console.ReadLine();  List<Library> list2 = libraryManager.AuthorSearch(str2);  if (list2.Count > 0)  {  foreach (var item in list2)  {  item.printInfo();  }  }  else  {  Console.WriteLine("Cannot find author!");  }  break;  case "6":  Console.WriteLine("Enter book's ID to be updated: ");  ID = int.Parse(Console.ReadLine());  libraryManager.Update(ID);  break;  case "7":  Console.WriteLine("Enter book's ID to be removed: ");  ID = int.Parse(Console.ReadLine());  libraryManager.RemoveBook(ID);  break;  }  } while (otp != "8");  }  }  } |

## **CHAPTER III: STEPS TO WRITING A PROGRAM (M1)**

# Understand the problem you are trying to solve

Go through the program, functions, features of the software that you are going to develop on thoroughly. Discuss with customers to find out the requirements for the program. Identify the end goal.

# Draw a flow chart

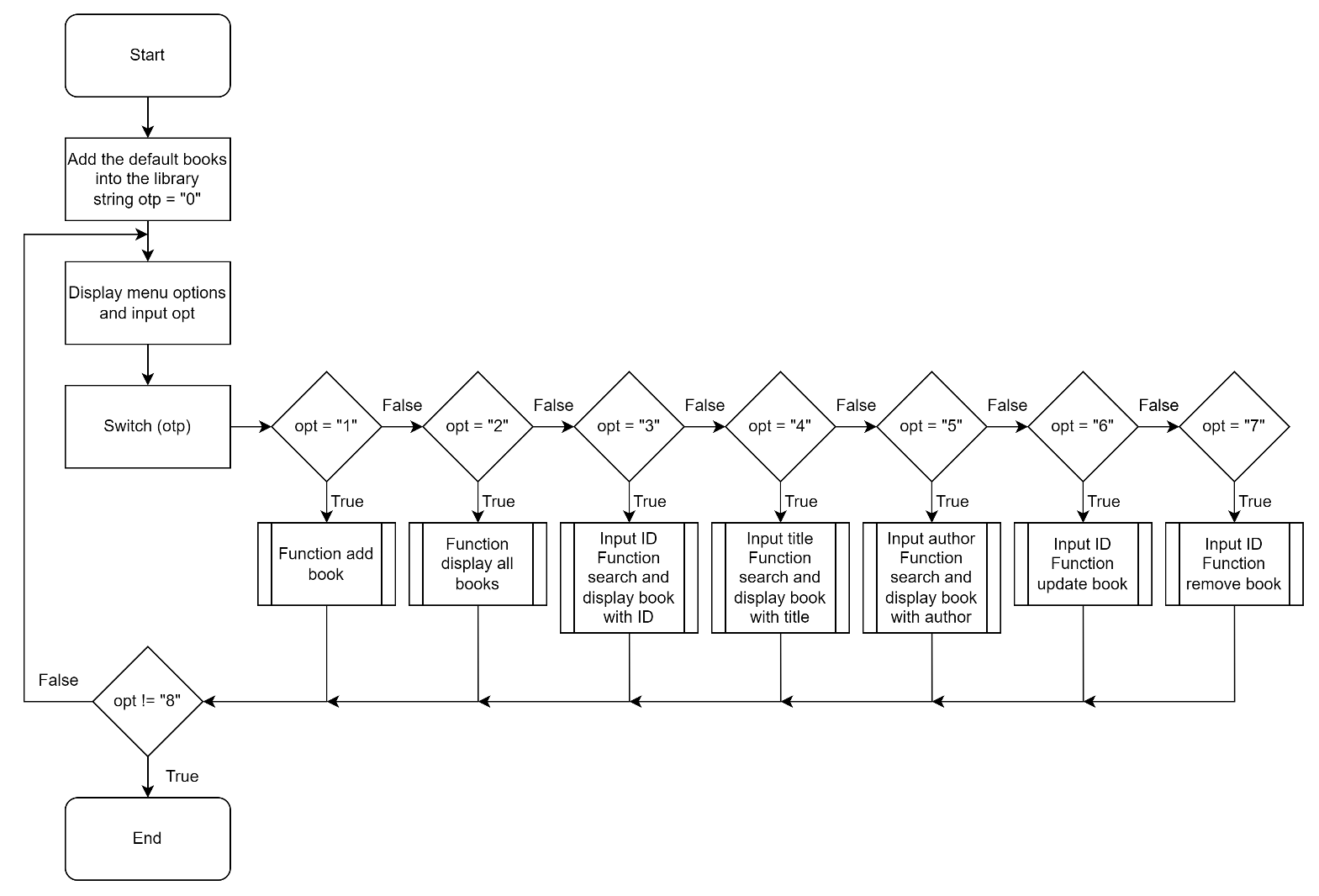
Make a flow chart or a UML for the problem at hand. Divide the problem into sub-problems or smaller chunks. Solve the subproblems. Make independent functions for each subproblem.

Connect the solutions of each subproblem by calling them in the required order, or as necessary.

Wherever it’s required use classes and objects while handling questions (for real-world problems like management systems, etc.)

For example:

Figure 13: MENU-BASE FLOWCHART



# Write pseudo-code

To make it easy to understand, pseudo-code isn’t a real code. Pseudo-code gives a clear idea that what your code is supposed to do by writing down every step and logic. Making it easier to write your code. In the end, you just need to replace pseudo-code with your code.

For example:

static void Main(string[] args)

{

Load the books into the library’s storage;

String otp = “0”;

Do

{

Display menu options and input string otp

Switch (otp)

{

Case “1”:

Function add book;

Break;

Case “2”:

Function display books;

Break;

Case “3”:

Input ID;

Function search and display book with ID;

Break;

Case “4”:

Input title;

Function search and display book with title;

Break;

Case “5”:

Input author;

Function search and display book with author;

Break;

Case “6”:

Input ID;

Function update book;

Break;

Case “7”:

Input ID;

Function remove book;

Break;

}

} while (otp != “8”)

}

# Replace pseudo-code with real code

Once you have written the pseudocode it’s time to translate this into actual code. Replace each line of your pseudocode into real code in the language you are working on. If you have divided your problem into subproblems then write down the code for each subproblem.

For example:

|  |
| --- |
| namespace AS1LibraryManagementSystem  {  internal class Program  {  static void Main(string[] args)  {  LibraryManager libraryManager = new LibraryManager();  libraryManager.BookInStorage();  string otp = "0";  do  {  otp = libraryManager.Menu();  switch (otp)  {  case "1":  libraryManager.AddBook();  break;  case "2":  libraryManager.displayBooks();  break;  case "3":  Console.WriteLine("Input ID to search (number): ");  string iD = Console.ReadLine();  while (string.Compare(iD, "0") == -1 || string.Compare(iD, "99999999999") == 1)  {  Console.Write("Wrong ID input. Input ID to search (number): ");  iD = Console.ReadLine();  }  int ID = int.Parse(iD);  Library id = libraryManager.IDSearch(ID);  if (id != null)  {  id.printInfo();  }  else  {  Console.WriteLine("Cannot find ID!");  }  break;  case "4":  Console.WriteLine("Input Title to search: ");  string title = Console.ReadLine();  List<Library> listOfTitle = libraryManager.TitleSearch(title);  if (listOfTitle.Count > 0)  {  foreach (var item in listOfTitle)  {  item.printInfo();  }  }  else  {  Console.WriteLine("Cannot find Title!");  }  break;  case "5":  Console.WriteLine("Input Author to search: ");  string author = Console.ReadLine();  List<Library> listOfAuthor = libraryManager.AuthorSearch(author);  if (listOfAuthor.Count > 0)  {  foreach (var item in listOfAuthor)  {  item.printInfo();  }  }  else  {  Console.WriteLine("Cannot find author!");  }  break;  case "6":  Console.WriteLine("Enter book's ID to be updated: ");  iD = Console.ReadLine();  while (string.Compare(iD, "0") == -1 || string.Compare(iD, "99999999999") == 1)  {  Console.Write("Wrong ID input. Input book's ID to update (number): ");  iD = Console.ReadLine();  }  ID = int.Parse(iD);  libraryManager.Update(ID);  break;  case "7":  Console.WriteLine("Enter book's ID to be removed: ");  iD = Console.ReadLine();  while (string.Compare(iD, "0") == -1 || string.Compare(iD, "99999999999") == 1)  {  Console.Write("Wrong ID input. Input book's ID to remove (number): ");  iD = Console.ReadLine();  }  ID = int.Parse(iD);  libraryManager.RemoveBook(ID);  break;  }  if (otp != "8")  {  Console.ReadKey();  Console.Clear();  }  } while (otp != "8");  }  }  } |

# Test and debug

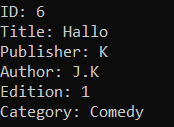
Always test your code with sample sets of data to check if the actual output is equal to the expected output.

* Test table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Object** | **Input data** | **Expected Output** | **Actual Output** | **Status** |
| 1 | Input a book | - Press 1 - Enter book’s ID: 6  - Enter book’s title: Hallo  - Enter book’s publisher: K  - Enter book’s author: J.K  - Enter book’s edition: 1  - Enter book’s category: Comedy | [success] | [success] | Pass |
| 2 | Display all books | - Press 2 | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | Pass |
| 3 | Search by ID | - Press 3  - Enter ID to Search: 1 | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | Pass |
| 4 | Search by title | - Press 4  - Enter title to search: Hallo | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | Pass |
| 5 | Search by author | - Press 5  - Enter author to search: J.K | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | ID: 6  Title: Hallo  Publisher: K  Author: J.K  Edition: 1  Category: Comedy | Pass |
| 6 | Update book | - Press 6  - Enter book’s ID to be updated: 6  - Input new publisher: Kim Dong  - Input new edition: 2  - Input new category: Horror  - Press 2 | ID: 6  Title: Hallo  Publisher: Kim Dong  Author: J.K  Edition: 2  Category: Horror | ID: 6  Title: Hallo  Publisher: Kim Dong  Author: J.K  Edition: 2  Category: Horror | Pass |
| 7 | Remove book | - Press 7  - Enter book’s ID to be removed: 1  - Press 2 | [null] | [null] | Pass |

* The console screen in testing:
  + Test-case 1:
    - Test object: Input a book
    - Status: Pass
    - Description: Input book’s descriptions into the library’s storage

Figure 14: TEST-CASE 1



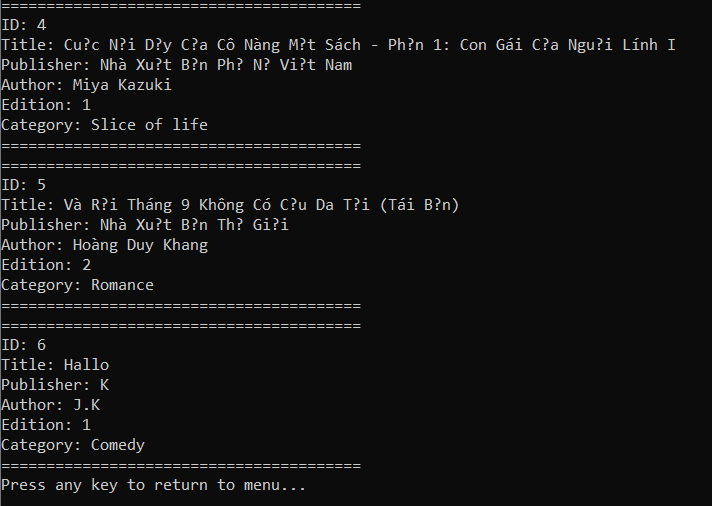
* + Test-case 2:
    - Test object: Display all book
    - Status: Pass
    - Description: Display all books that is stored in the library’s storage
  + Test-case 3:

Figure 15: TEST-CASE 2

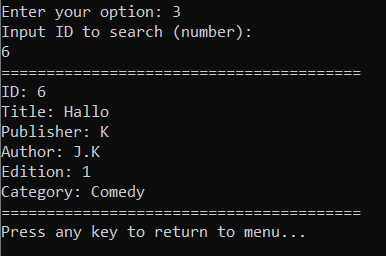
* + - Test object: Search by ID
    - Status: Pass
    - Description: Search and display the book with the ID inputted
  + Test-case 4:

Figure 16: TEST-CASE 3

* + - Test object: Search by title
    - Status: Pass
    - Description: Search and display the book with the inputted title

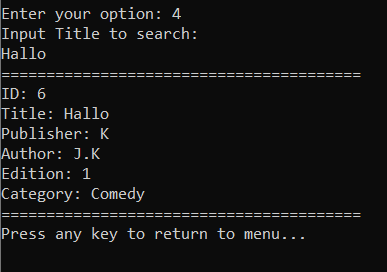


Figure 17: TEST-CASE 4

* + Test-case 5:
    - Test object: Search by author
    - Status: Pass
    - Description: Search and display the book with the inputted author

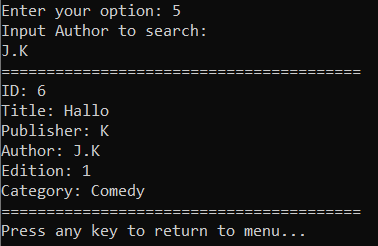


Figure 18: test-case 5

* + Test-case 6:
    - Test object: Update book
    - Status: Pass
    - Description: Search and update the book with the inputted ID

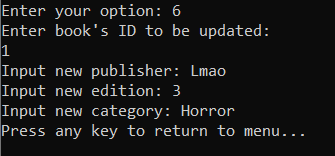
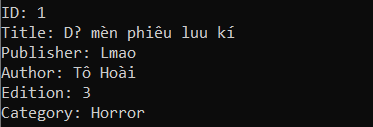
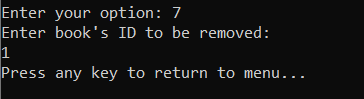


Figure 19: test-case 6

* + Test-case 7:
    - Test object: Remove book
    - Status: Pass
    - Description: Search and remove the book with the inputted ID

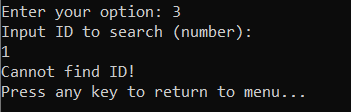


Figure 20: test-case 7

## **CHAPTER IV: EVALUATION (D1)**

* Evaluate the relationship between code and algorithm:

Algorithm and code are different, but related things. The relationship is simple: Code expresses algorithms.

The same algorithm could be expressed in different languages, including natural language, some form of pseudocode, machine language and programming languages. Barring natural language and pseudocode, we refer to these representations as code (source code if it is not machine language).

In fact, translating from some languages to others is possible and done routinely from programming languages to machine language, or from a programming language to another. Sometimes from machine language back to a programming language. This evidences that the same algorithm can be expressed in multiple languages, which results in different but equivalent code.

I'm saying that code is more concrete that algorithms. Algorithms are ideas, and we can use code to portray them, study them, compare them, talk about them. Or have a computer execute them.

Algorithms can exist before being portrayed (in code, for example). That is to say, you can figure out how to do something before writing it down. It is the expression that makes it code. (Mortensen, 2022)

## **REFERENCES**

Mortensen, P., 2022. *Difference between Algorithm and Code*. [online] Software Engineering Stack Exchange. Available at: <https://softwareengineering.stackexchange.com/questions/423254/difference-between-algorithm-and-code#:~:text=The%20relationship%20is%20simple%3A%20Code%20expresses%20algorithms.%20The,%28source%20code%20if%20it%20is%20not%20machine%20language%29.> [Accessed 15 August 2022].

S. Gillis, A., 2022. *What is an Algorithm? - Definition from WhatIs.com*. [online] WhatIs.com. Available at: <https://www.techtarget.com/whatis/definition/algorithm> [Accessed 15 August 2022].

Codesansar. 2022. *Algorithms (Characteristics, Guidelines & Advantages)*. [online] Available at: <https://www.codesansar.com/computer-basics/algorithms.htm> [Accessed 14 August 2022].

ABDULWADUD SOUBRA, M., 2022. *Program Development Life Cycle in 6 Steps*. [online] Mql5.com. Available at: <https://www.mql5.com/en/blogs/post/713281> [Accessed 14 August 2022].