ENGG1110 — Problem Solving by Programming
2019–2020 Term 1
Project Specification — A Vending Machine Simulator

1. Introduction

Vending machines are automated machines that provide different products to consumers after payment is accepted. For simplicity, we will be focusing on vending machines that take coins (available in Hong Kong) and provide beverages. A functional vending machine should also allow service personnel to collect revenue and refill inventory.

In this project, your task is to use C language to write a vending machine simulator that mimics its real counterpart.

2. Functionalities

Similar to a real vending machine, the simulator should provide the following functions.

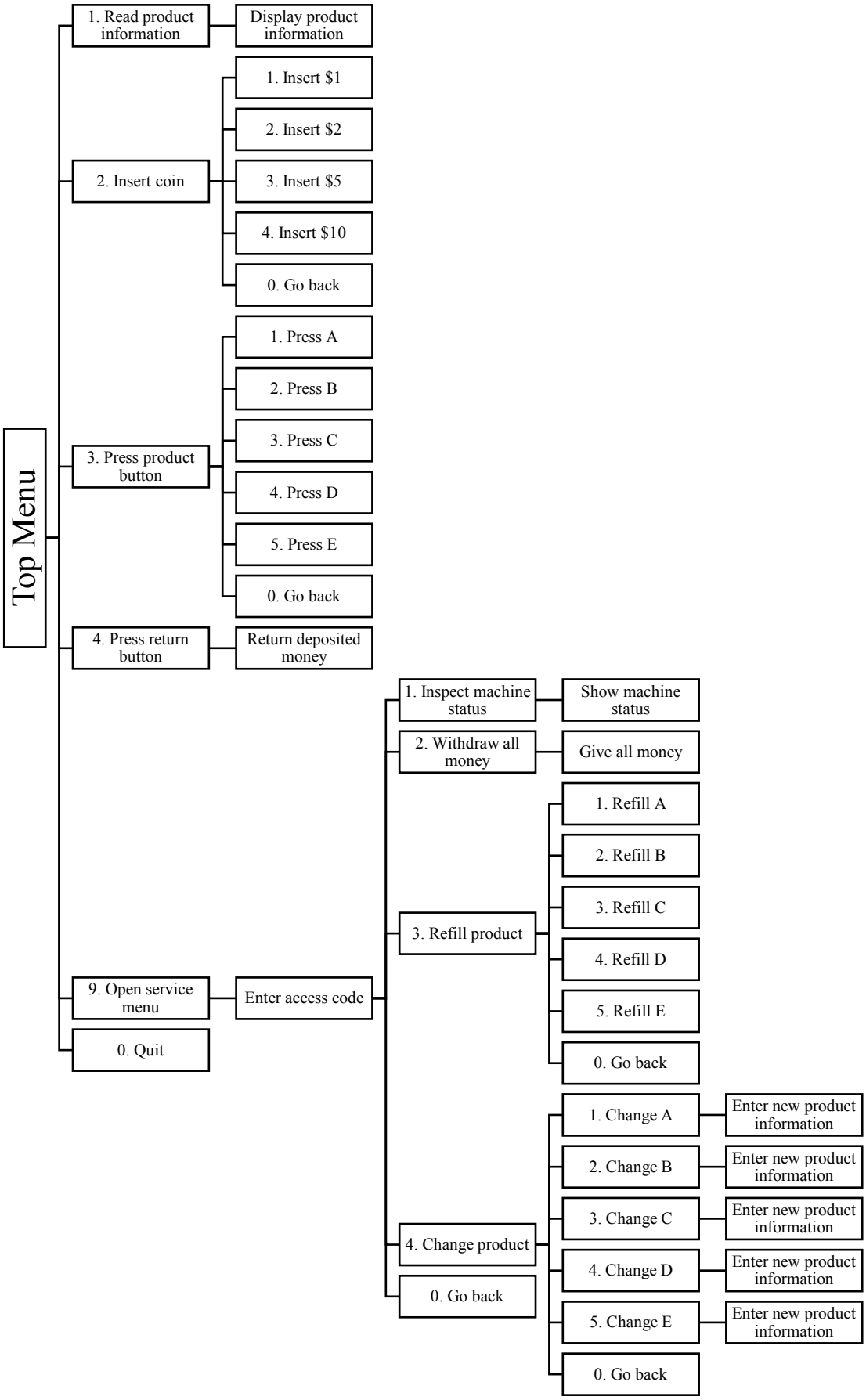
For normal consumers:

(a) Display product names and prices
(b) Indicate if a product is out-of-stock
(c) Take inserted coins
(d) Indicate if inserted coin is sufficient for a product
(e) Dispense product
(f) Return inserted coins and change

For service personnel:

(g) Allow withdrawal of all inserted coins
(h) Allow a product to be refilled
(i) Allow a different product to be displayed

3. Menu Map


Top Menu
1. Read product information
Display product information
2. Insert coin
1. Insert $1
2. Insert $2
3. Insert $5
4. Insert $10
0. Go back
3. Press product button
1. Press A
2. Press B
3. Press C
4. Press D
5. Press E
0. Go back
4. Press return button
Return deposited money
9. Open service menu
Enter access code
1. Inspect machine status
Show machine status
2. Withdraw all money
Give all money
3. Refill product
1. Refill A
2. Refill B
3. Refill C
4. Refill D
5. Refill E
0. Go back
4. Change product
1. Change A
Enter new product information
2. Change B
Enter new product information
3. Change C
Enter new product information
4. Change D
Enter new product information
5. Change E
Enter new product information
0. Go back
0. Go back
0. Quit

4. Detailed Program Flow

When the program starts, it should print the following front of the vending machine. Then a main menu with available options should be shown.

```
*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7 |
|  [ ]  [ ]  [ ]  [ ]  [ ] |
*--------------------------*
|                    [$ 0] |
|                          |
|           [===]          |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

Whenever an invalid choice is entered for **any menu**, the program should report and keep looping. For example (**input is underlined**):

```
What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice: 8
Invalid choice!

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

(1) Read product information

If option 1 is chosen, the product names and the corresponding price will be displayed. Then the main menu is displayed again.

```
What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice: 1

(1) The displayed products are:
A. Juice ($10)
B. Cola ($6)
C. Tea ($5)
D. Water ($8)
E. Coffee ($7)

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

(2) Insert coin

If option 2 is chosen, a sub-menu with available coin options should be shown. After a coin is inserted, the front of vending machine with correct value of deposited coins should be shown (maximum: $99), then the sub-menu is shown again.

(Assume the machine has $0 deposited at the beginning.)

```
(2) Which coin would you like to insert?
1. $1
2. $2
3. $5
4. $10
0. Go back
Your choice: 2
You have inserted $2.

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7 |
|  [ ]  [ ]  [ ]  [ ]  [ ] |
*--------------------------*
|                    [$ 2] |
|                          |
|           [===]          |
*--------------------------*

(2) Which coin would you like to insert?
1. $1
2. $2
3. $5
4. $10
0. Go back
Your choice:
```

If the deposited money is sufficient to purchase a product, indicate it by “turning on the light” using a capital “O”.

(Continuing from previous step.)

```
(2) Which coin would you like to insert?
1. $1
2. $2
3. $5
4. $10
0. Go back
Your choice: 3
You have inserted $5.

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E   |
|  $10  $ 6  $ 5  $ 8  $ 7  |
|  [ ]  [O]  [O]  [ ]  [O]  |
*--------------------------*
|                    [$ 7]  |
|                           |
|           [===]           |
*--------------------------*

(2) Which coin would you like to insert?
1. $1
2. $2
3. $5
4. $10
0. Go back
Your choice:
```

Selecting option 0 (Go back) will show the vending machine, and then return to the main menu.

(Continuing from previous step.)

```
(2) Which coin would you like to insert?
1. $1
2. $2
3. $5
4. $10
0. Go back
Your choice: 0
Going back!

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7 |
|  [ ]  [O]  [O]  [ ]  [O] |
*--------------------------*
|                    [$ 7] |
|                          |
|           [===]          |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

(3) Press product button

If option 3 is chosen, a sub-menu with available product options should be shown. If the deposited money is sufficient to purchase the product chosen, the product will be dispensed. The front of vending machine with correct product dispensed, button lights and deposited value should be shown, then the main menu is shown.

(Assume that the machine has $7 deposited, and product B, C and E are affordable and available. And there is only one product B left.)

```
(3) Which product button would you like to press?
1. A
2. B
3. C
4. D
5. E
0. Go back
Your choice: 2
You have pressed button B.

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7 |
|  [ ]  [X]  [ ]  [ ]  [ ] |
*--------------------------*
|                    [$ 1] |
|                          |
|           [=B=]          |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

Note that product B has just been sold out. There is an “X” on its button now. Also note that the remaining deposited amount will not be returned automatically. The product dispensed is assumed to be taken when available, so that future display will not show this dispensed product again.

If a button for an out-of-stock product or an unaffordable product is pressed, nothing will happen.

(Continuing from the setup of the previous step, but starting from the Press Product Button sub-menu.)

```
(3) Which product button would you like to press?
1. A
2. B
3. C
4. D
5. E
0. Go back
Your choice: 1
You have pressed button A.

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7  |
|  [ ]  [X]  [ ]  [ ]  [ ]  |
*--------------------------*
|                    [$ 1]  |
|                           |
|           [===]           |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

Selecting option 0 (Go back) will show the vending machine, and then return to the main menu (same as before; not shown).

(4) Press return button

If option 4 is chosen, all the deposited money will be returned. The front of vending machine with correct information should be shown, then the main menu is shown again.

(Continuing from previous step.)

```
(4) Return button is pressed.
$1 has been returned.

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 7  |
|  [ ]  [X]  [ ]  [ ]  [ ]  |
*--------------------------*
|                    [$ 0]  |
|                           |
|           [===]           |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

(9) Service menu

If option 9 is chosen, the user is required to enter an access code (default: 1110). Incorrect code will cause the simulator to return to the front display and the main menu after showing an error message.

```
(9) Opening service menu. Access code is required.
Enter access code: 1234
Incorrect code!

*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E   |
|  $10  $ 6  $ 5  $ 8  $ 7  |
|  [ ]  [X]  [ ]  [ ]  [ ]  |
*--------------------------*
|                    [$ 0]  |
|                           |
|           [===]           |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice:
```

Correct access code will allow the service menu to be shown.

```
(9) Opening service menu. Access code is required.
Enter access code: 1110
Correct code!

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice:
```

(9-1) Inspect machine status

If option 1 is chosen in the service menu, the information about the revenue (money comes from sales), inserted coins (money inserted but not used) and the products will be displayed. Then the service menu is shown again.

```
(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice: 1

(9-1) Machine status
Amount of revenue: $6
Amount of inserted coins: $0
Product information:
A. Juice ($10) (5 left)
B. Cola ($6) (sold out)
C. Tea ($5) (2 left)
D. Water ($8) (1 left)
E. Coffee ($7) (9 left)

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice:
```

(9-2) Withdraw all money

If option 2 is chosen in the service menu, all money in the machine (including coins that are inserted but not used) will be withdrawn. Then the service menu is shown again.

(Continuing from previous step.)

```
(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice: 2

(9-2) All money is being withdrawn.
$6 is withdrawn.

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice: 2

(9-2) All money is being withdrawn.
$0 is withdrawn.

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice:
```

(9-3) Refill product

If option 3 is chosen in the service menu, a sub-menu will be shown for the user to select the product to refill to the full quantity (fixed at 10 items). Then the service menu will be shown again.

```
(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice: 3

(9-3) Which product would you like to refill?
1. A
2. B
3. C
4. D
5. E
0. Go back
Your choice: 2
You have refilled product B to full.

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice:
```

(9-4) Change product

If option 4 is chosen in the service menu, a sub-menu will be shown for the user to select the product to change.

```
(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice: 4

(9-4) Which product would you like to change?
1. A
2. B
3. C
4. D
5. E
0. Go back
Your choice: 5
You are changing product E.
```

(9-4-1) Changing product

After the product has been selected, the user is prompted to enter a new product name (maximum 20 characters with no space) and new product price ($1–$99 with one dollar increments). The quantity will be filled to full. Then the service menu is shown again.

```
Enter new product name: Milk
Enter new product price: 4
The new product E has been filled to full.

(9) What would you like to do?
1. Inspect machine status
2. Withdraw all money
3. Refill product
4. Change product
0. Go back
Your choice:
```

(0) Quit

If option 0 is chosen in the top menu, the program will terminate. No message is given.

```
*--------------------------*
|      Vending Machine     |
*--------------------------*
|   A    B    C    D    E  |
|  $10  $ 6  $ 5  $ 8  $ 4  |
|  [ ]  [ ]  [ ]  [ ]  [ ]  |
*--------------------------*
|                    [$ 0]  |
|                           |
|           [===]           |
*--------------------------*

What would you like to do?
1. Read product information
2. Insert coin
3. Press product button
4. Press return button
9. Open service menu (code required)
0. Quit
Your choice: 0

>
```

5. Academic Honesty and Declaration Statement

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details may be found at https://www.cuhk.edu.hk/policy/academichonesty/.

Please put the following declaration statement as the comment in the beginning of your source code and fill in your information.

```
/**
* ENGG1110 Problem Solving by Programming
*
* Course Project
*
* I declare that the project here submitted is original
* except for source material explicitly acknowledged,
* and that the same or closely related material has not been
* previously submitted for another course.
* I also acknowledge that I am aware of University policy and
* regulations on honesty in academic work, and of the disciplinary
* guidelines and procedures applicable to breaches of such
* policy and regulations, as contained in the website.
*
* University Guideline on Academic Honesty:
* https://www.cuhk.edu.hk/policy/academichonesty/
*
* Student Name : <your name>
* Student ID : <your student ID>
* Class/Section : <your class/section>
* Date : <date>
*/
```

6. Testing Platform

Your code will be tested on Repl.it platform. Modular testing will be provided to let you confirm the correctness of certain parts of your program.

7. Code and Report Submission

Your code will be submitted on **both** Repl.it **and** Blackboard. The submission links will be available at a later date, and you will be notified when they are opened.

You are also required to submit a flow chart describing the flow of your program.

- Flow charts are expected to be as detail as in the lecture notes (see p. 17–19 of 10. Logic Flow).
- The submitted flow chart should be in JPG, PDF, DOC/DOCX, or PPT/PPTX formats. The submission links will be available at a later date, and you will be notified when they are opened.

8. Schedule

Submission (code and flowchart) by **23:59, Sunday, December 15, 2019** on Blackboard

9. Assumptions and Initial Conditions

You can safely assume the following conditions during program execution.

(1) All menu inputs from user are integers.
(2) All product names have maximum 20 characters and no spaces in between.
(3) All product prices are between $1 and $99. When single-digit money is displayed on the machine (prices and inserted coins), it is shown as "`$ 0`". When double-digit money is displayed on the machine, it is shown as "`$10`". When displayed in messages, there is no space after "`$`" in both cases.
(4) Maximum amount of inserted coins is $99. Behavior is not defined if more than $99 are inserted.
(5) Access code for service menu (option 9) will be entered as integer, and will be treated as integer. So both entered codes **1110** and **001110** will grant access.
(6) When changing product (action 9-4-1), newly entered product name and price will obey the above assumptions (2) and (3).
(7) The newly entered product name and price will be shown in Product Information (in Main menu) and Inspect Machine Status (in Service menu), until the program terminates.
(8) No need preserve machine status (i.e., revenue, inserted coins, product names and prices) between program executions. Every time the program starts, it should be reset to the initial conditions (as described below).

Your vending machine should have the following initial conditions every time the program starts.

- Amount of revenue: $0
- Amount of inserted coins: $0
- Product information:
    A. Juice ($10) (5 left)
    B. Cola ($6) (1 left)
    C. Tea ($5) (2 left)
    D. Water ($8) (1 left)
    E. Coffee ($7) (9 left)