UNIVERSITY
OF WOLLONGONG
IN DUBAI

Coffee Machine Interface in C

Muthana Al-Sirhan, #8785995

University of Wollongong in Dubai
CSCI291, Lab #2 Report

Table of Contents

Lab Objective, Methodology 3
Main Menu 4
Ordering Coffee (Flowchart) 5
Ordering Coffee (Order Process) 6
Ordering Coffee (Payment Process) 7
Admin Mode (Flowchart) 8
Admin Mode (Display Quantities) 9
Admin Mode (Replenishing Ingredients) 10
Admin Mode (Changing Prices) 11
Admin Mode (Reset & Collect Sales) 12

1. Lab Objective, Methodology

Self service Coffee Machines are a commodity in all kinds of areas including universities, convenience stores and outside in general. The aim of this lab is to program an interface for a hypothetical self-service Coffee Machine that can serve cups of Espresso, Cappuccino and Mocha, in the C programming language. It should allow customers to order their coffee cup(s) of choice and its operators to display/replenish ingredients, collect total sales and change the price for the aforementioned coffee types. There are a few design limitations applied to the code, such as the inability to implement any arrays, pointers or C-structs.

To implement all this functionality into the C program, constants are defined after:-

- The price and ingredient requirements for a cup of each coffee type (Table 1)
- The Low Threshold Quantity for each ingredient in the machine. When an ingredient passes that threshold, a message is displayed instructing the customer to contact an on-site machine operator.
- A real-value variable which gives the total sales amount, **`salesAED`** in this case.

Then, functions are prototyped and defined to act as the front-end and back-end of the interface, examples including:-

- Menus embedded within infinite while loops, allowing users to navigate the interface
- Ordering coffee and inserting money
- An Admin Mode (password required) to display/replenish coffee ingredients and change coffee price.

Table 1: Coffee Type Cup: Ingredients & Prices

Coffee Type	Coffee Beans (g)	Water (mL)	Milk (mL)	Choc. Syrup (mL)	Price (AED)
Espresso	8	30	-	-	3.5
Cappuccino	8	30	70	-	4.5
Mocha	8	39	160	30	5.5

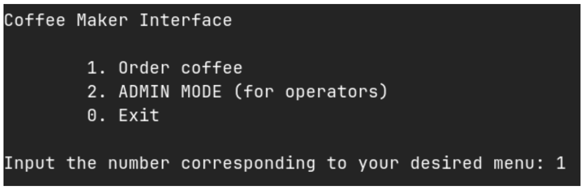
Flowcharts have been designed after the aforementioned functions to streamline the programming process (included further in this report).

2. Main Menu [main()]

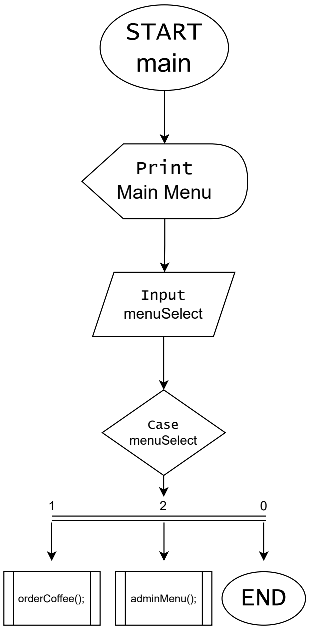
The main function in this code displays (or "prints") the main menu interface embedded within an infinite while loop. Then, it scans the users integer input corresponding to their desired menu choice. The input is passed to a switch statement with the following cases:

- 1. Coffee Ordering Menu [calls orderCoffee()]
- 2. Admin Mode [calls adminMode()]
- 0. Quitting the Program [returns 0 to the inf. loop]

And a default case for any other input, which prints "INVALID OPTION".

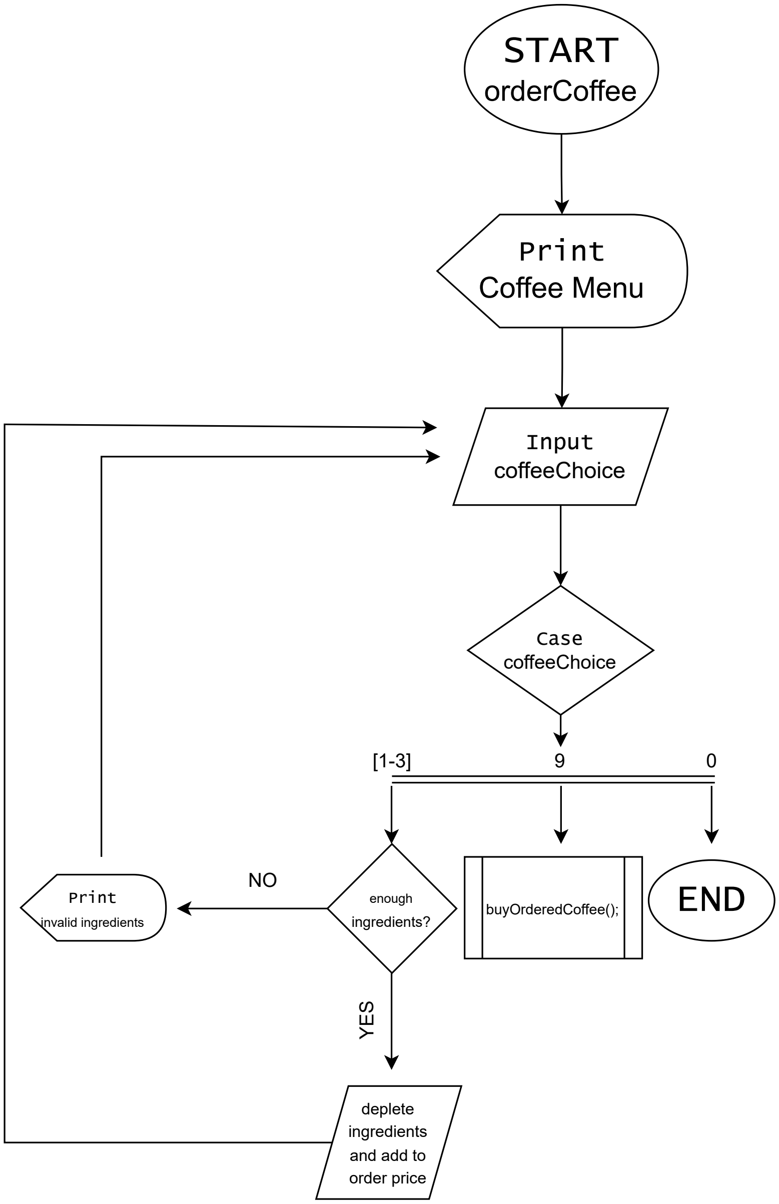


```
Coffee Maker Interface

        1. Order coffee
        2. ADMIN MODE (for operators)
        0. Exit

Input the number corresponding to your desired menu: 1
```



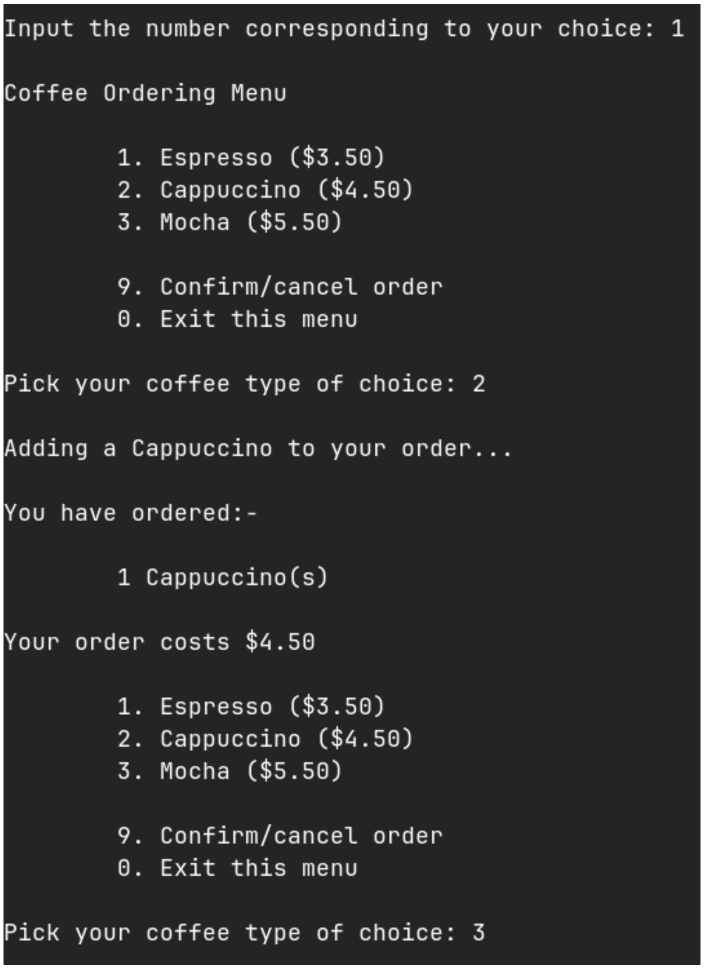
3. Ordering Coffee [orderCoffee()]

Customers can order coffee through the menu invoked by the orderCoffee() function. During the process, a small list of the final order is always displayed to the customer along with the final price.



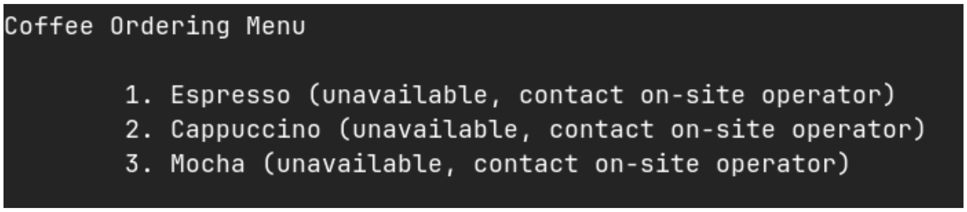
3.1. Order process

Each cup ordered calls a function that depletes ingredients and adds to the total order price, depending on the type of coffee ordered.



```
Input the number corresponding to your choice: 1

Coffee Ordering Menu

        1. Espresso ($3.50)
        2. Cappuccino ($4.50)
        3. Mocha ($5.50)

        9. Confirm/cancel order
        0. Exit this menu

Pick your coffee type of choice: 2

Adding a Cappuccino to your order...

You have ordered:-

        1 Cappuccino(s)

Your order costs $4.50

        1. Espresso ($3.50)
        2. Cappuccino ($4.50)
        3. Mocha ($5.50)

        9. Confirm/cancel order
        0. Exit this menu

Pick your coffee type of choice: 3
```

Conditional statements are implemented to check if an ingredient(s) has passed the low threshold amount and to print a message instructing to contact an on-site machine operator to replenish the ingredients.



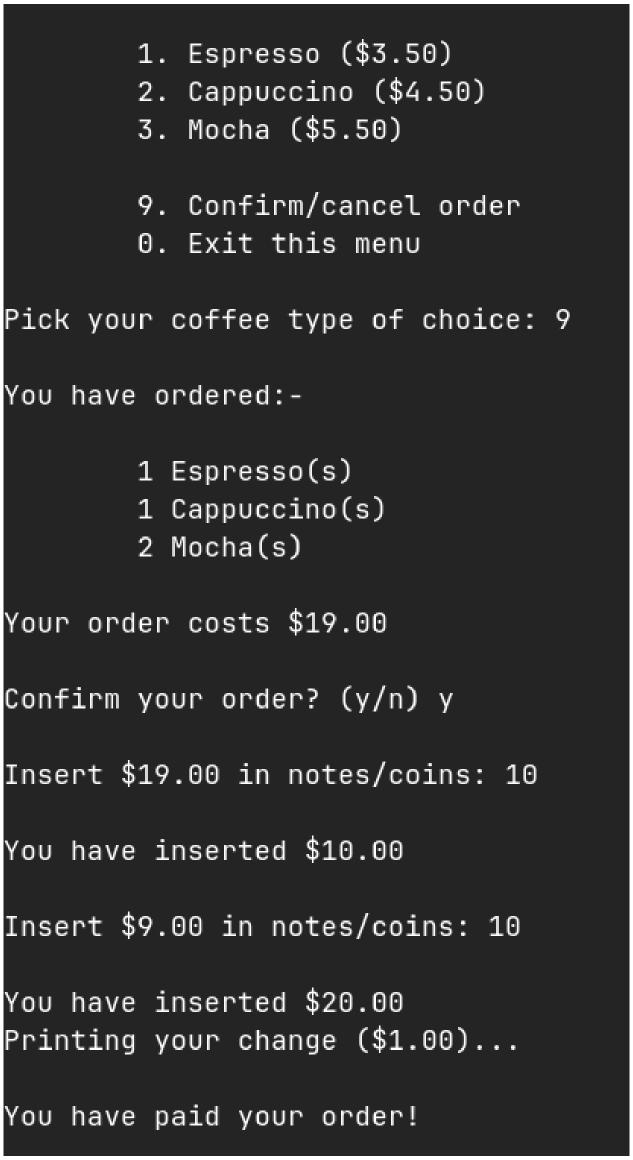
```
Coffee Ordering Menu

        1. Espresso (unavailable, contact on-site operator)
        2. Cappuccino (unavailable, contact on-site operator)
        3. Mocha (unavailable, contact on-site operator)
```

3.2. Payment process

When a customer has finished and confirmed the order, it can be paid with:-

- 20, 10, 5 AED notes
- 1, 0.5, 0.25 AED coins

The remaining change is deposited by the machine.

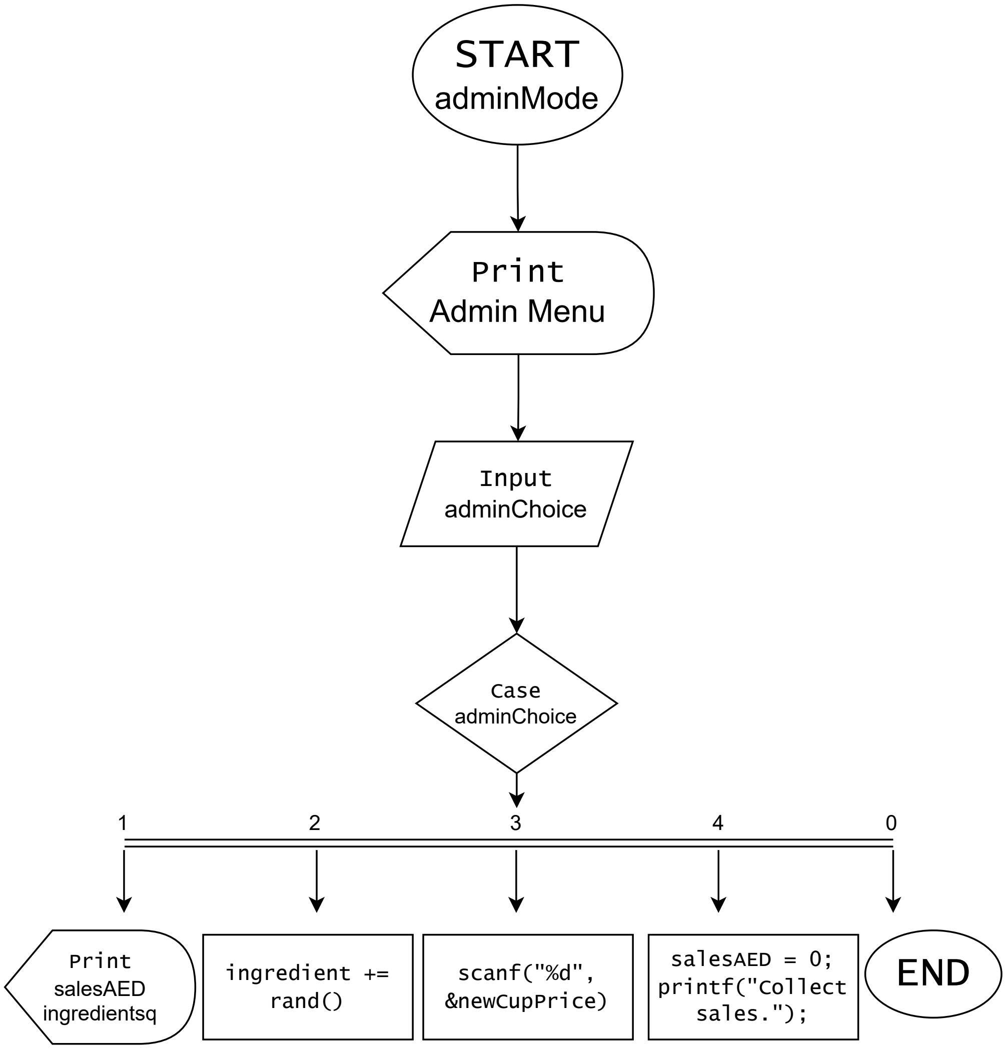


```
        1. Espresso ($3.50)
        2. Cappuccino ($4.50)
        3. Mocha ($5.50)

        9. Confirm/cancel order
        0. Exit this menu

Pick your coffee type of choice: 9

You have ordered:-

        1 Espresso(s)
        1 Cappuccino(s)
        2 Mocha(s)

Your order costs $19.00

Confirm your order? (y/n) y

Insert $19.00 in notes/coins: 10

You have inserted $10.00

Insert $9.00 in notes/coins: 10

You have inserted $20.00
Printing your change ($1.00)...

You have paid your order!
```

4. Operating The Machine [adminMode()]

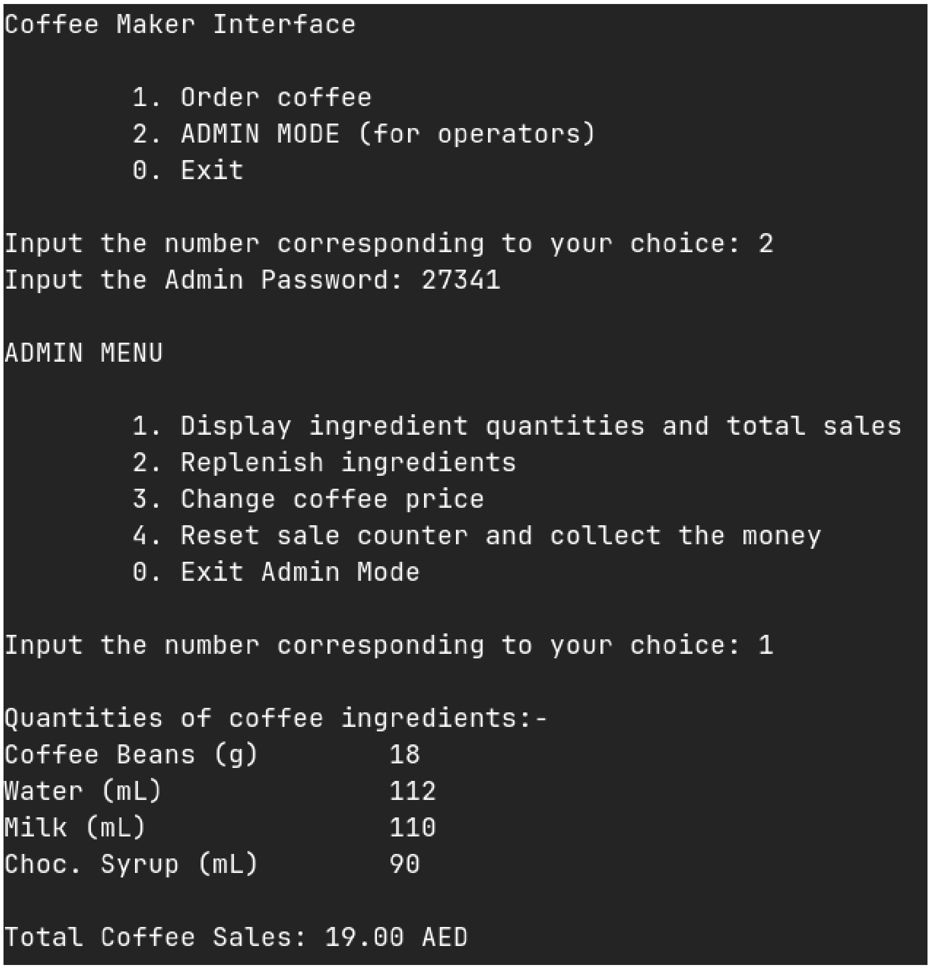
Operators can log into the interface’s "Admin Mode" with the password (27341 in this case). In Admin Mode, various functionality is provided to the operator as follows:-

- Displaying ingredient quantities and total sales
- Replenishing ingredients
- Changing the price of a coffee cup
- Resetting and collecting sales



4.1. Display all quantities

Simply prints the quantities of all the ingredients alongside **`salesAED`**

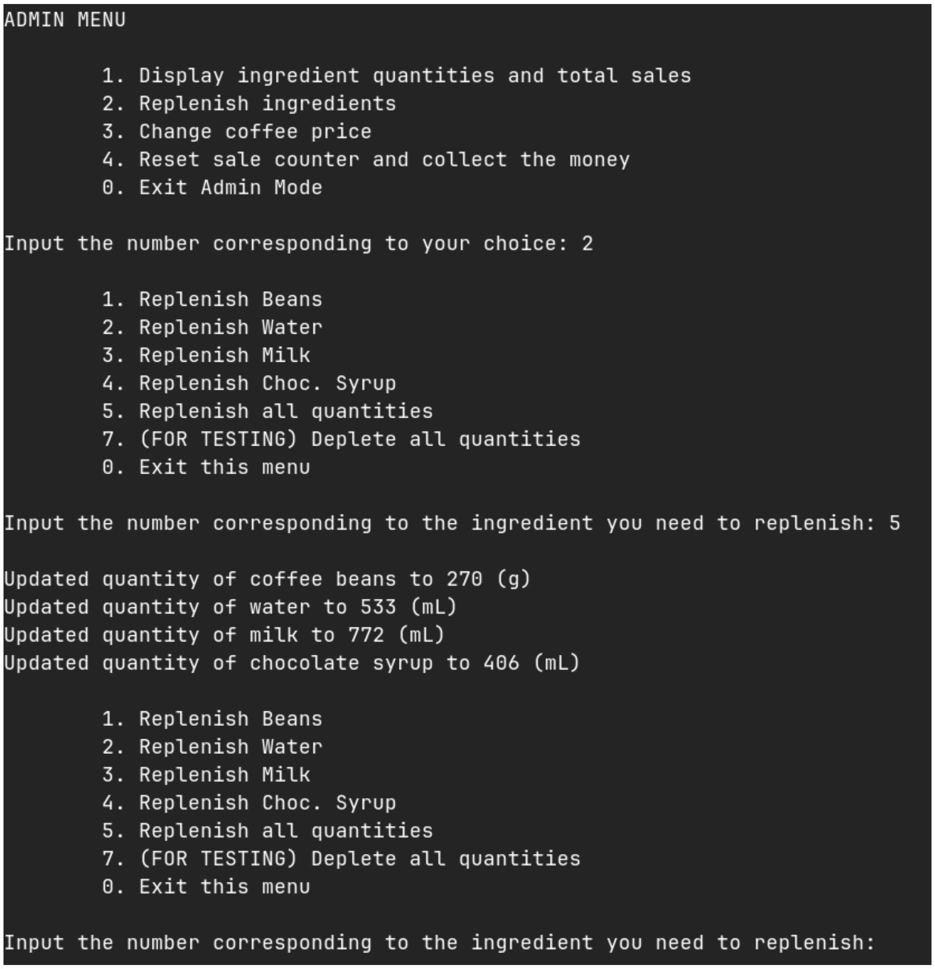


```
Coffee Maker Interface

        1. Order coffee
        2. ADMIN MODE (for operators)
        0. Exit

Input the number corresponding to your choice: 2
Input the Admin Password: 27341

ADMIN MENU

        1. Display ingredient quantities and total sales
        2. Replenish ingredients
        3. Change coffee price
        4. Reset sale counter and collect the money
        0. Exit Admin Mode

Input the number corresponding to your choice: 1

Quantities of coffee ingredients:-
Coffee Beans (g)        18
Water (mL)              112
Milk (mL)               110
Choc. Syrup (mL)        90

Total Coffee Sales: 19.00 AED
```

4.2. Replenishing Ingredients

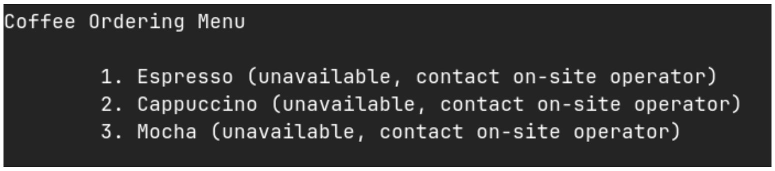
With the use of the **rand()** function provided by the <stdlib.h> header, random integers from 0 to 32,767 can be generated [1]. To ensure that it generates reasonable numbers for our coffee ingredients, we can specify a range [min, max] with the following "formula":-

```
ingredient = (rand() % (max-min)) + min
```



```
ADMIN MENU

        1. Display ingredient quantities and total sales
        2. Replenish ingredients
        3. Change coffee price
        4. Reset sale counter and collect the money
        0. Exit Admin Mode

Input the number corresponding to your choice: 2

        1. Replenish Beans
        2. Replenish Water
        3. Replenish Milk
        4. Replenish Choc. Syrup
        5. Replenish all quantities
        7. (FOR TESTING) Deplete all quantities
        0. Exit this menu

Input the number corresponding to the ingredient you need to replenish: 5

Updated quantity of coffee beans to 270 (g)
Updated quantity of water to 533 (mL)
Updated quantity of milk to 772 (mL)
Updated quantity of chocolate syrup to 406 (mL)

        1. Replenish Beans
        2. Replenish Water
        3. Replenish Milk
        4. Replenish Choc. Syrup
        5. Replenish all quantities
        7. (FOR TESTING) Deplete all quantities
        0. Exit this menu

Input the number corresponding to the ingredient you need to replenish:
```

One case for resetting the ingredient variables to 0 was implemented for quick testing to ensure no orders can be placed when there are insufficient ingredients.

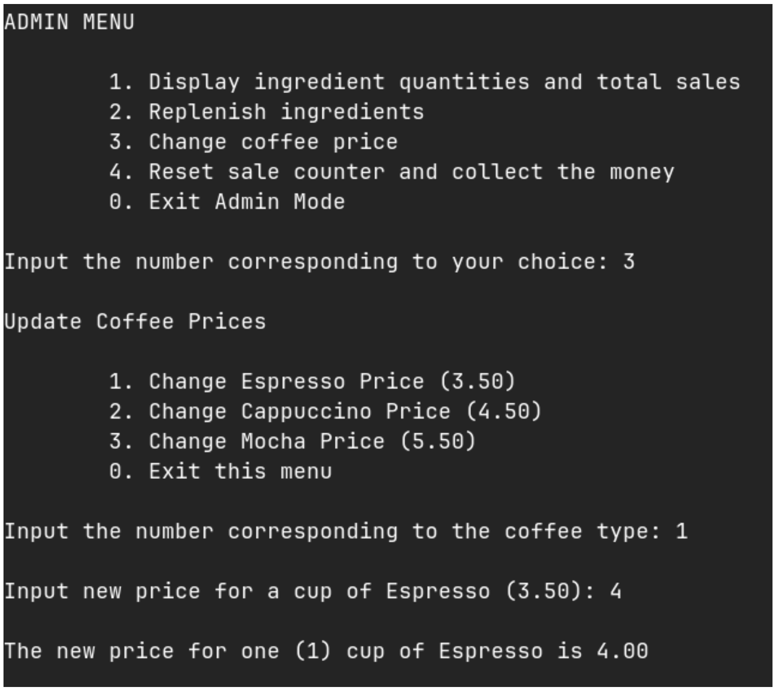


```
Coffee Ordering Menu

        1. Espresso (unavailable, contact on-site operator)
        2. Cappuccino (unavailable, contact on-site operator)
        3. Mocha (unavailable, contact on-site operator)
```

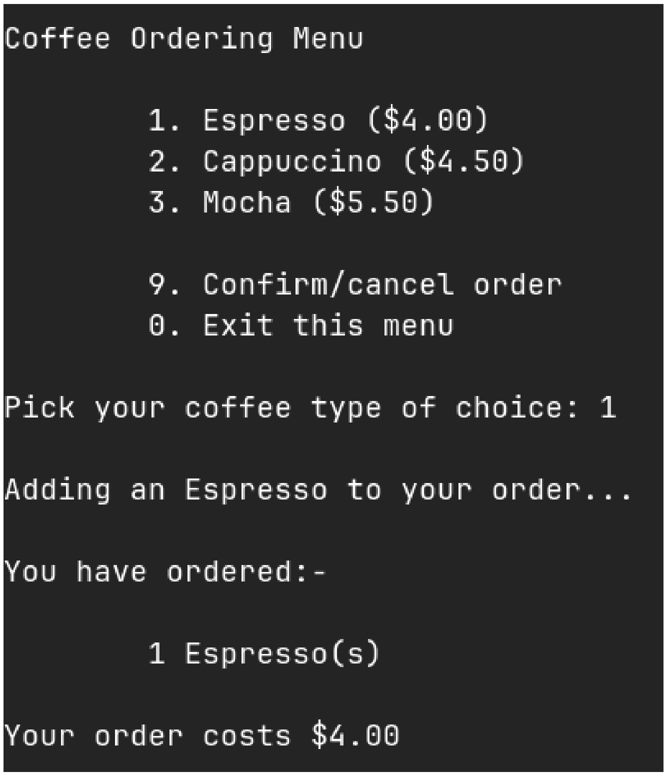
[1] https://mathbits.com/MathBits/CompSci/LibraryFunc/rand.htm

4.3. Changing Coffee Price

The operator is prompted to input a new price for the chosen cup. Only positive floats/integers are allowed, otherwise the new price isn’t implemented.



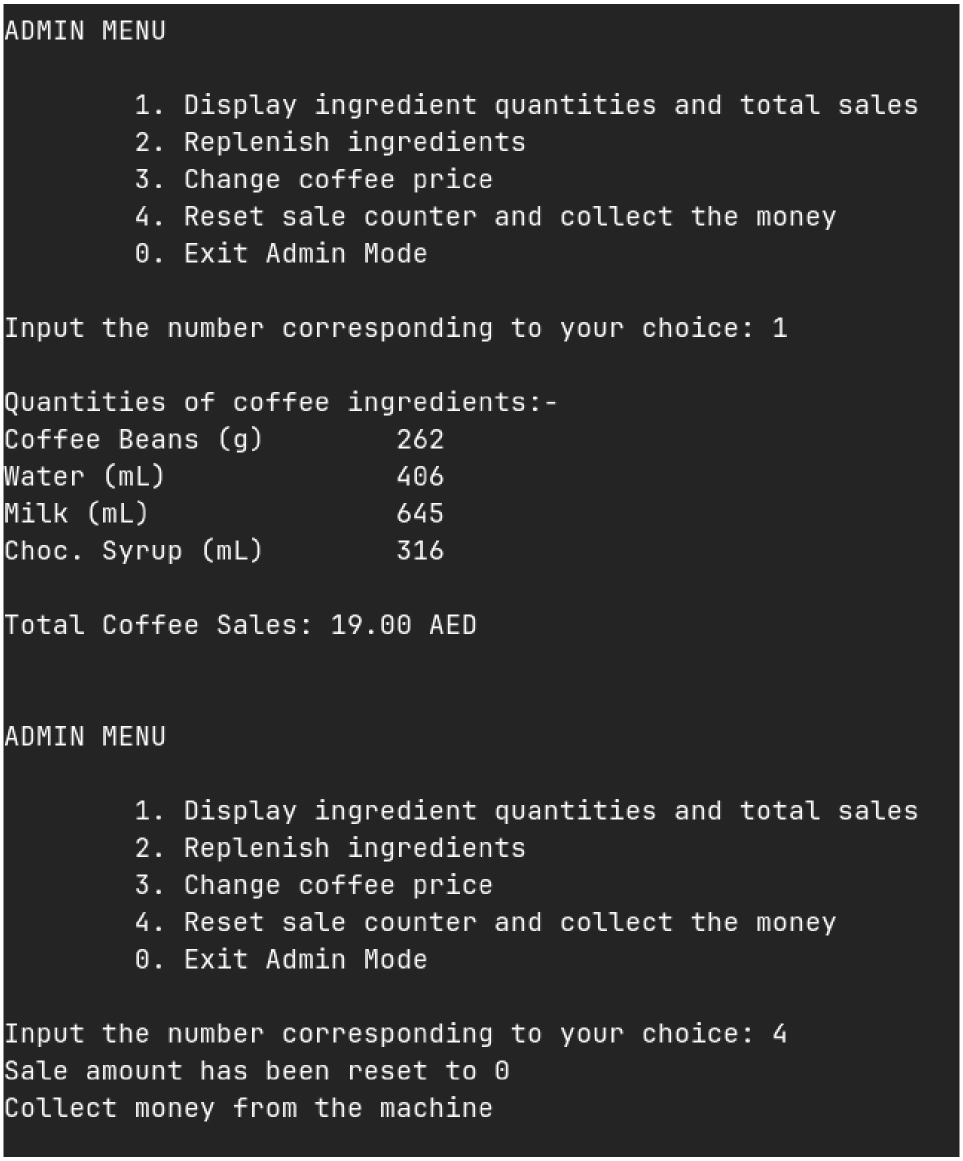
```
ADMIN MENU

        1. Display ingredient quantities and total sales
        2. Replenish ingredients
        3. Change coffee price
        4. Reset sale counter and collect the money
        0. Exit Admin Mode

Input the number corresponding to your choice: 3

Update Coffee Prices

        1. Change Espresso Price (3.50)
        2. Change Cappuccino Price (4.50)
        3. Change Mocha Price (5.50)
        0. Exit this menu

Input the number corresponding to the coffee type: 1

Input new price for a cup of Espresso (3.50): 4

The new price for one (1) cup of Espresso is 4.00
```



```
Coffee Ordering Menu

        1. Espresso ($4.00)
        2. Cappuccino ($4.50)
        3. Mocha ($5.50)

        9. Confirm/cancel order
        0. Exit this menu

Pick your coffee type of choice: 1

Adding an Espresso to your order...

You have ordered:-

        1 Espresso(s)

Your order costs $4.00
```

4.4. Reset & Collect Sales

The operator can reset the total sale variable by the end of a business day and collect the sales, dispensed by the machine.



```
ADMIN MENU

        1. Display ingredient quantities and total sales
        2. Replenish ingredients
        3. Change coffee price
        4. Reset sale counter and collect the money
        0. Exit Admin Mode

Input the number corresponding to your choice: 1

Quantities of coffee ingredients:-
Coffee Beans (g)        262
Water (mL)              406
Milk (mL)               645
Choc. Syrup (mL)        316

Total Coffee Sales: 19.00 AED


ADMIN MENU

        1. Display ingredient quantities and total sales
        2. Replenish ingredients
        3. Change coffee price
        4. Reset sale counter and collect the money
        0. Exit Admin Mode

Input the number corresponding to your choice: 4
Sale amount has been reset to 0
Collect money from the machine
```