CPT111 Principles of Programming

Academic session 2020/21

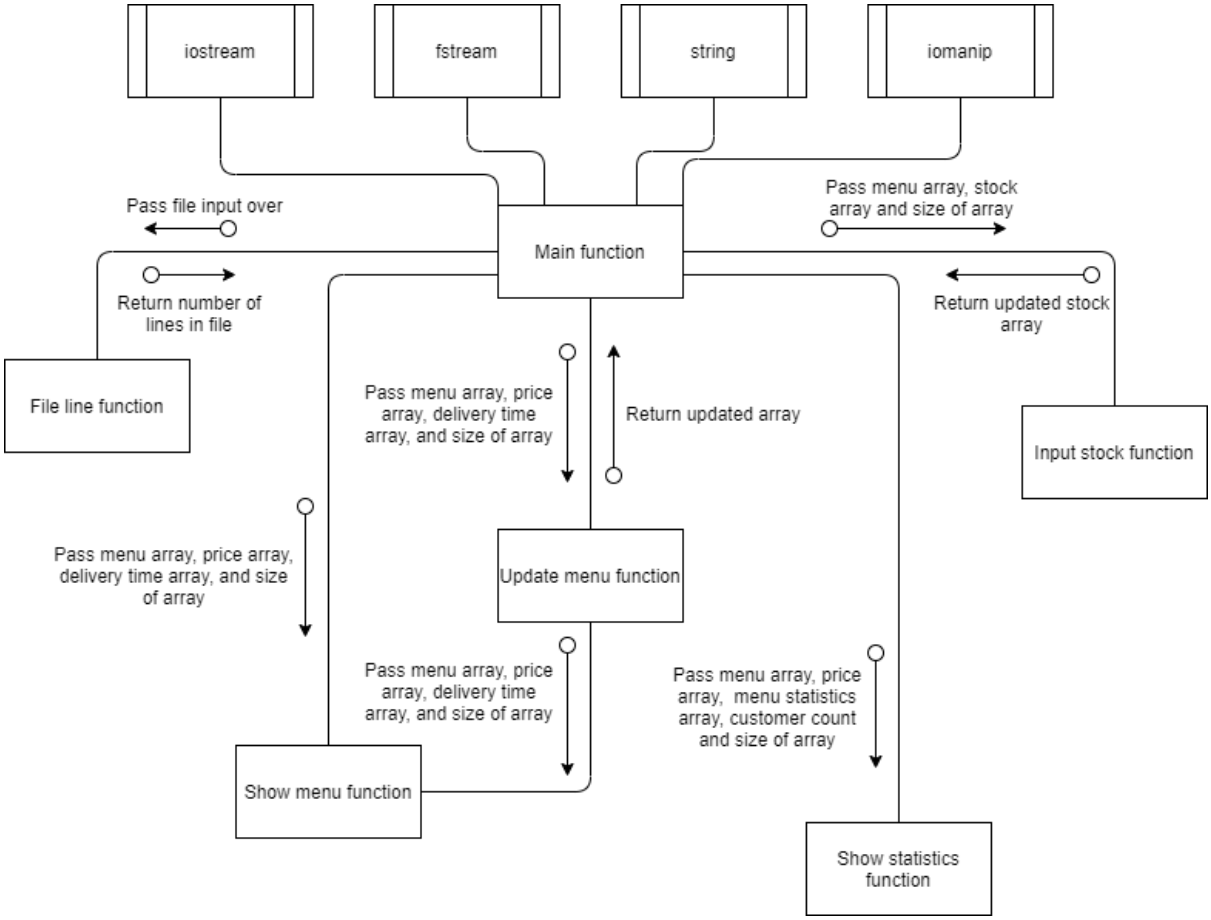
Assignment 2

RITCHIE POH ritchiepoh@student.usm.my **153765**

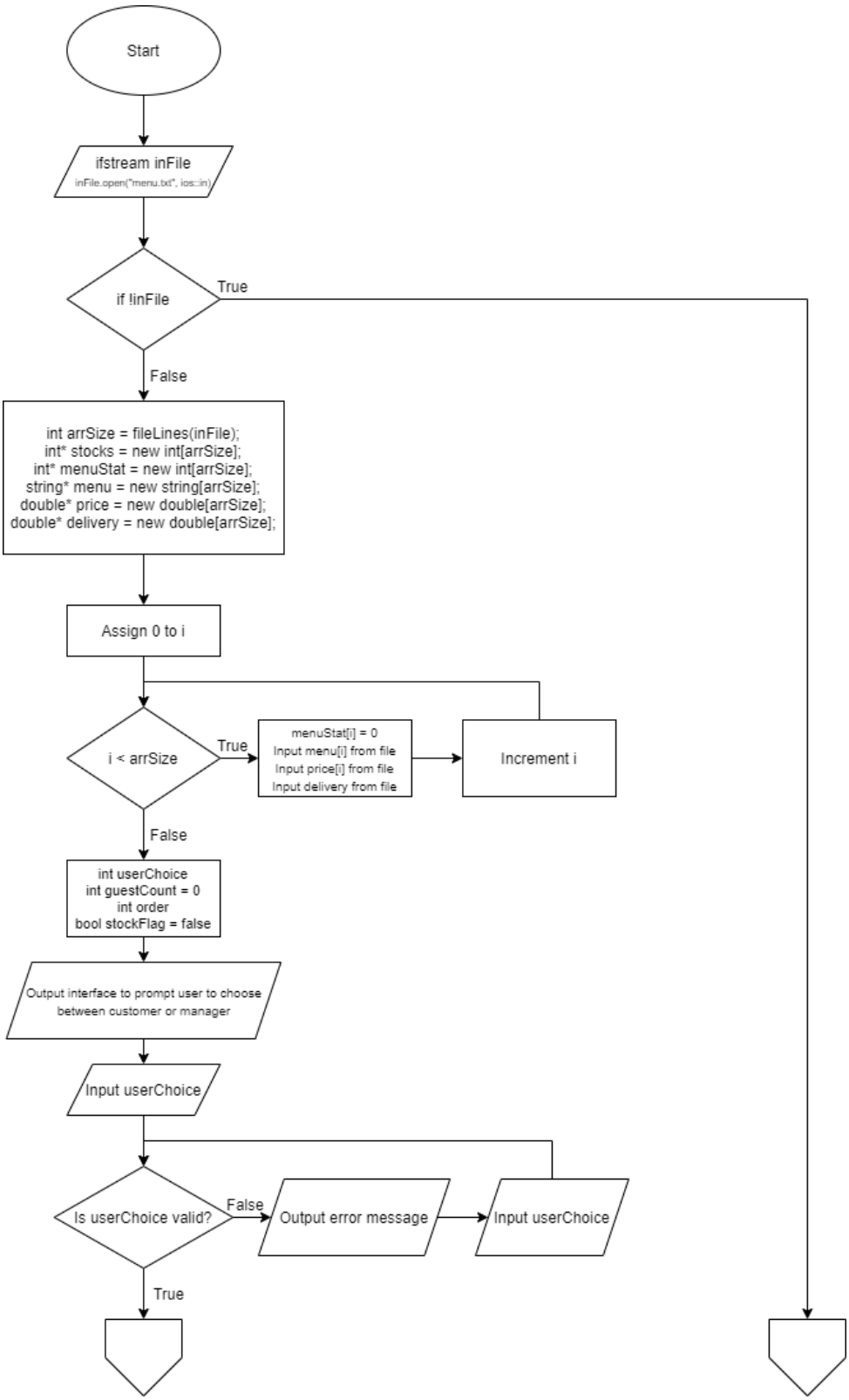
Table of contents

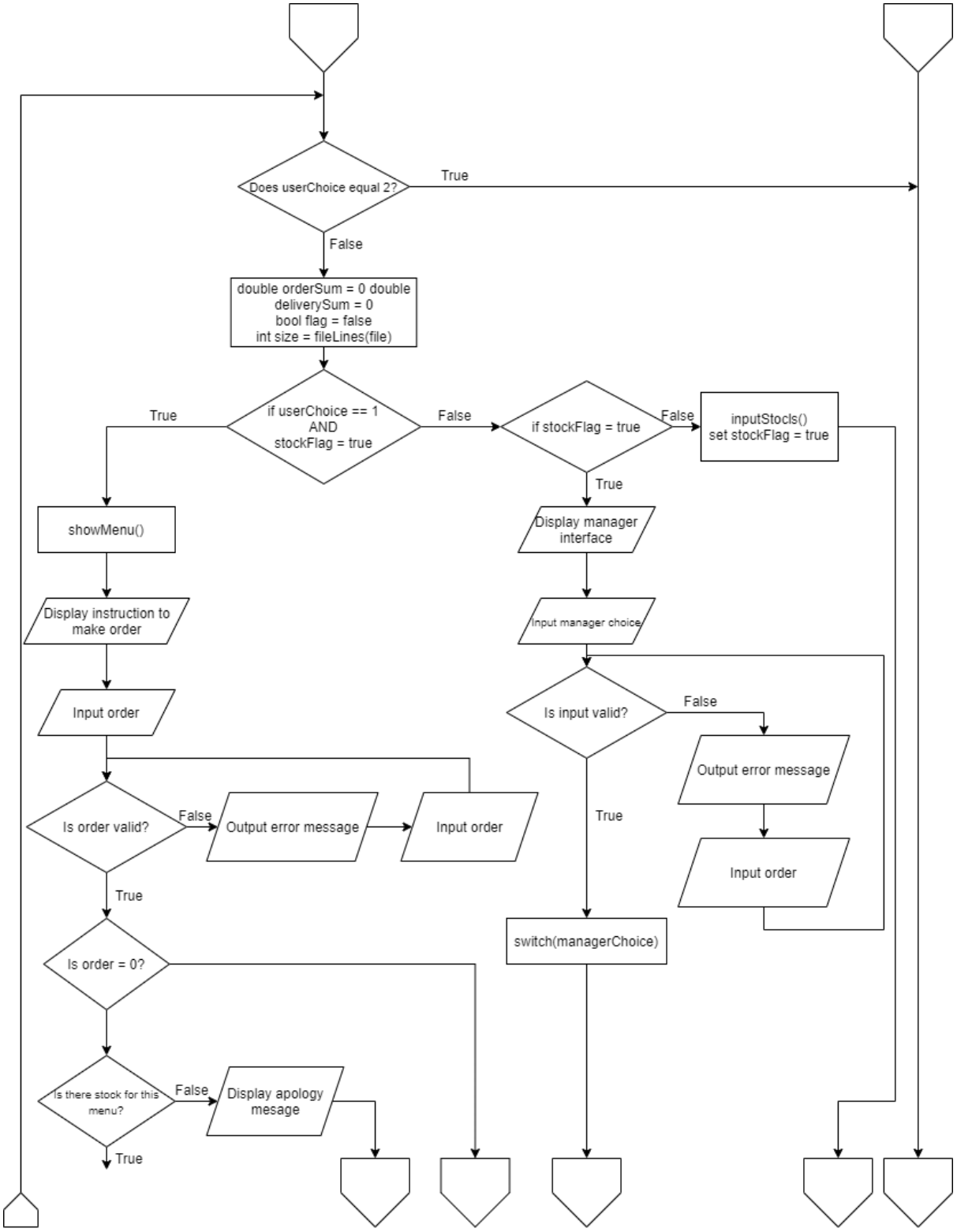
- **Structure chart** **Pg 2**
- **Flow charts** **Pg 3**
- **Program listings** **Pg 13**
- **Screenshots** **Pg 28**

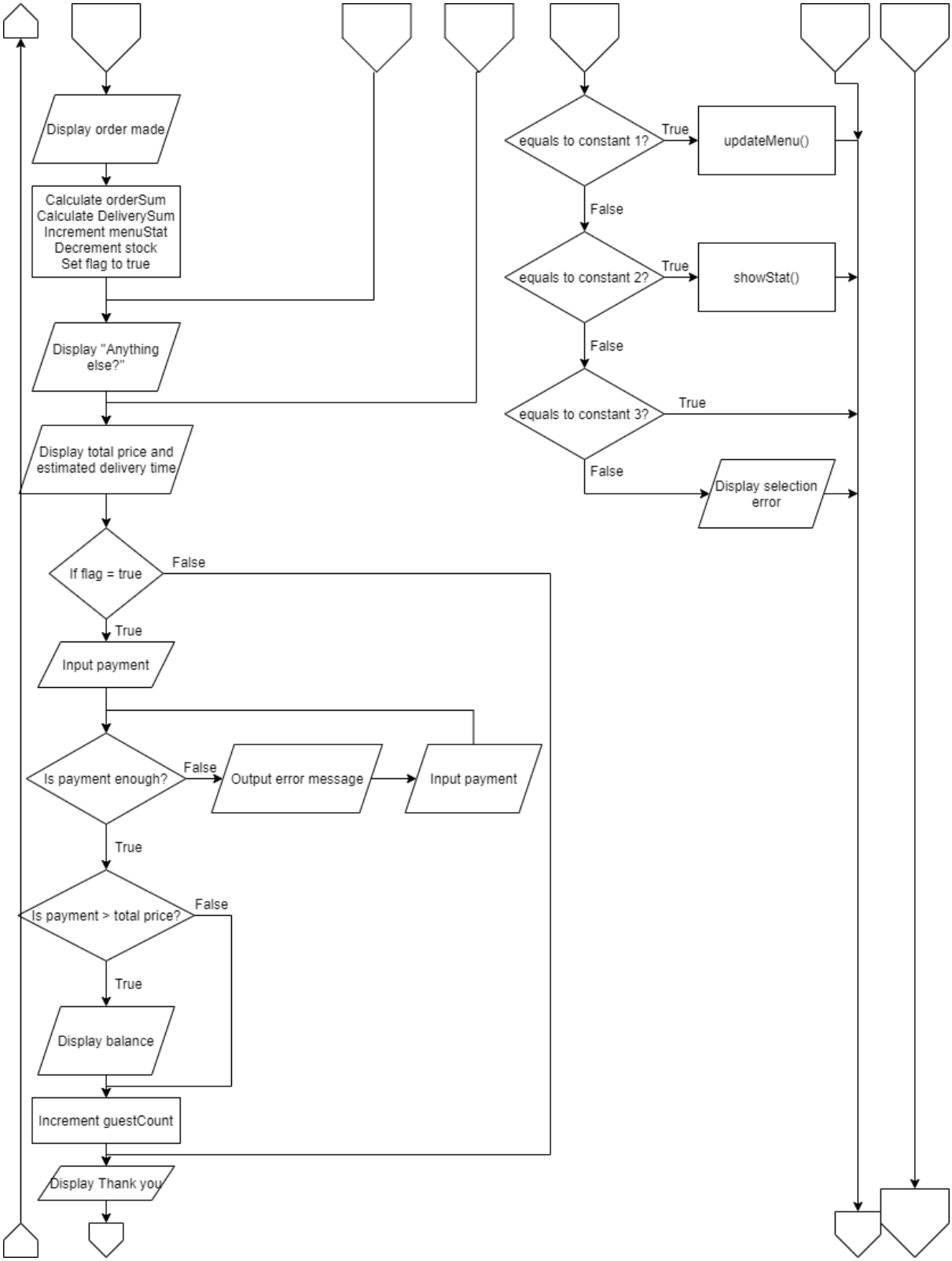
Structure Chart:

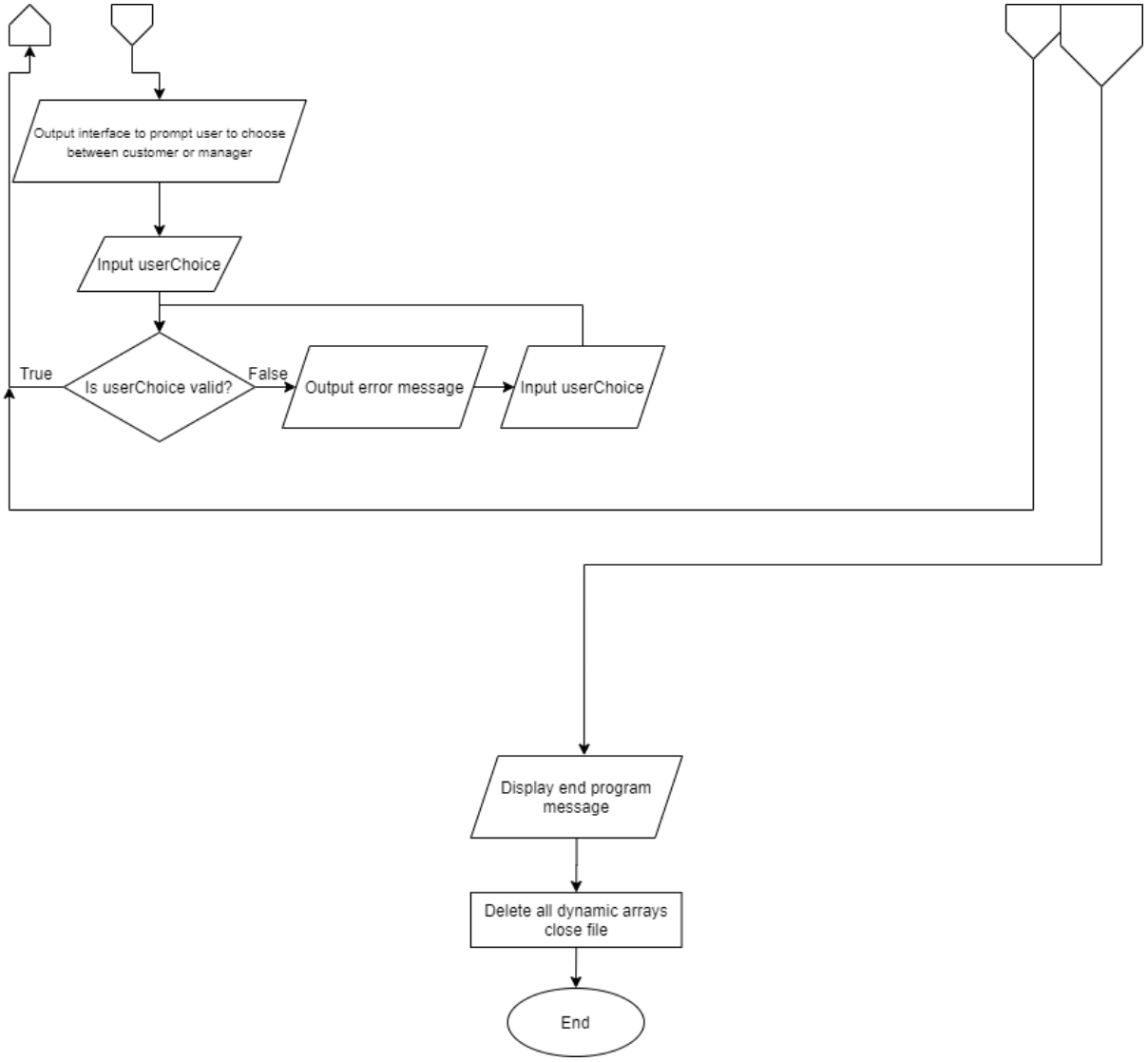


Flow chart of main function:

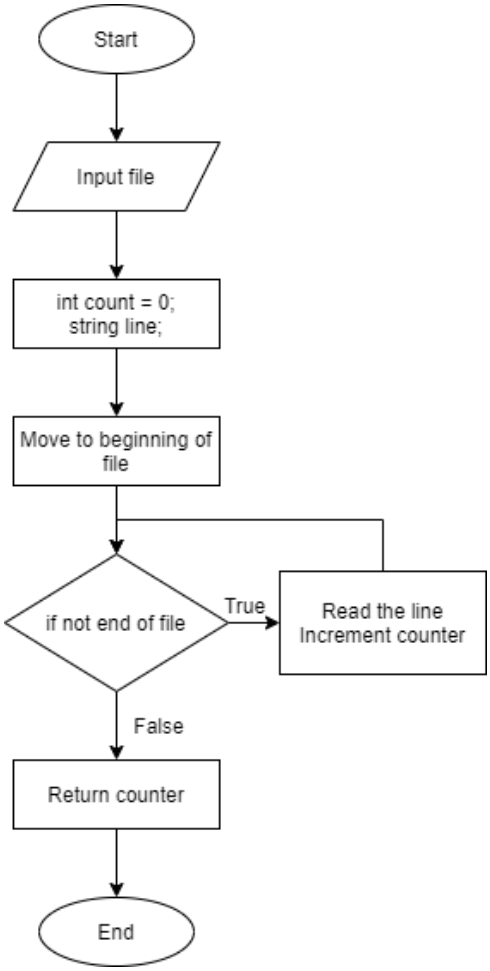



Does userChoice equal 2?
True
False
double orderSum = 0 double
deliverySum = 0
bool flag = false
int size = fileLines(file)
if userChoice == 1
AND
stockFlag = true
True
False
if stockFlag = true
False
inputStocls()
set stockFlag = true
True
showMenu()
Display manager
interface
Display instruction to
make order
Input manager choice
Input order
Is input valid?
False
Output error message
Is order valid?
False
Output error message
Input order
True
Input order
True
switch(managerChoice)
Is order = 0?
Is there stock for this
menu?
False
Display apology
mesage
True

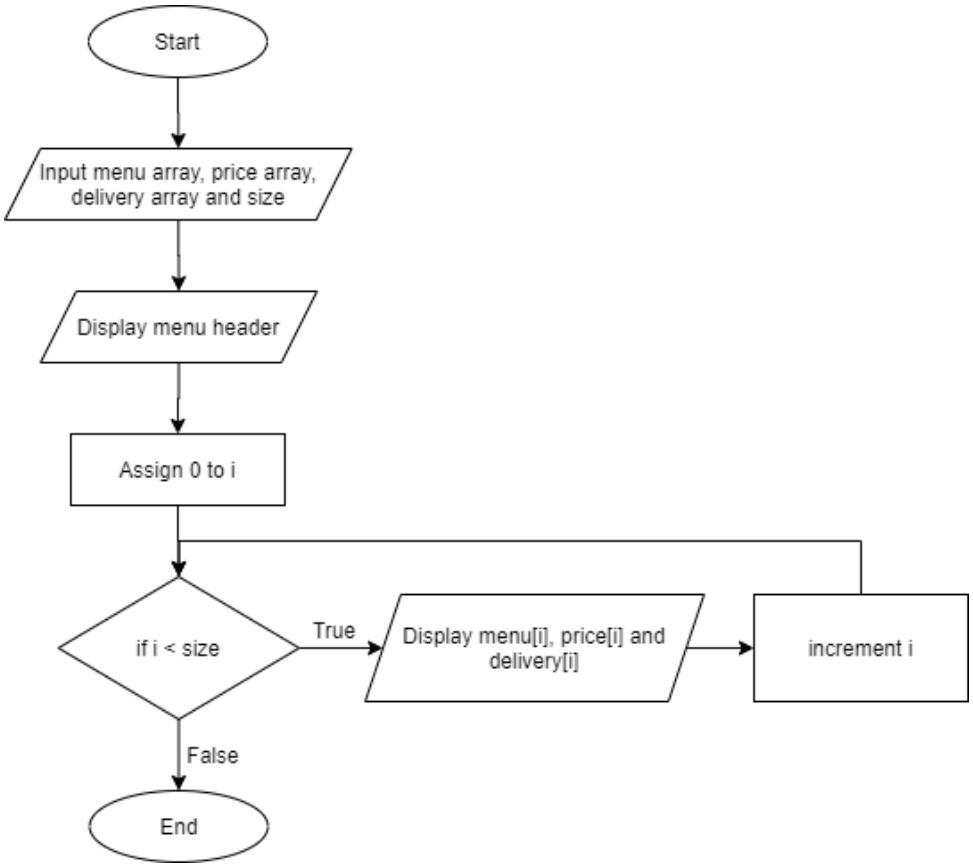

Display order made
Calculate orderSum
Calculate DeliverySum
Increment menuStat
Decrement stock
Set flag to true
Display "Anything else?"
Display total price and estimated delivery time
If flag = true
False
True
Input payment
Is payment enough?
False
Output error message
Input payment
True
Is payment > total price?
False
True
Display balance
Increment guestCount
Display Thank you
equals to constant 1?
True
updateMenu()
False
equals to constant 2?
True
showStat()
False
equals to constant 3?
True
False
Display selection error


Output interface to prompt user to choose between customer or manager
Input userChoice
True
Is userChoice valid?
False
Output error message
Input userChoice
Display end program message
Delete all dynamic arrays close file
End

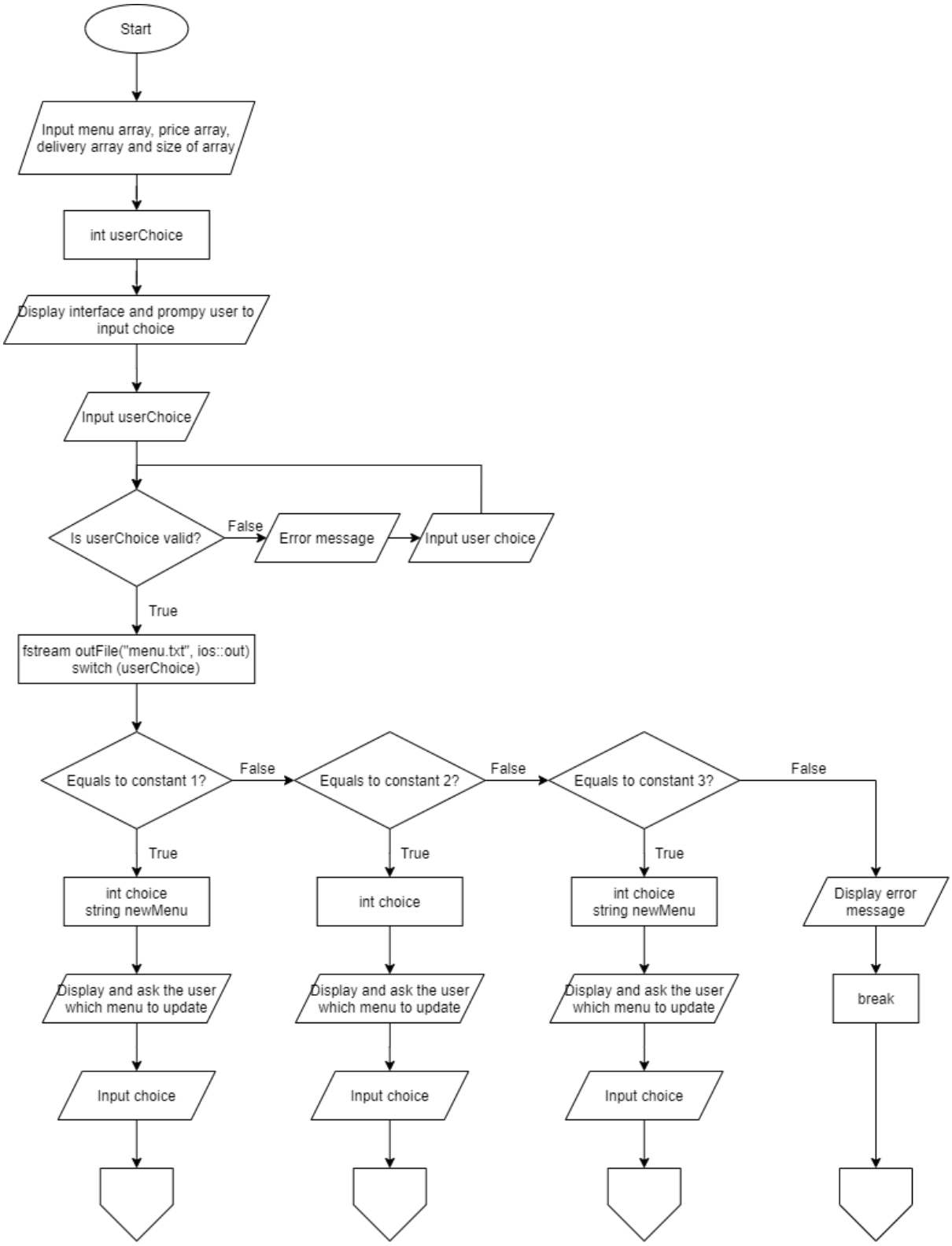
Flow chart for fileLine() function:

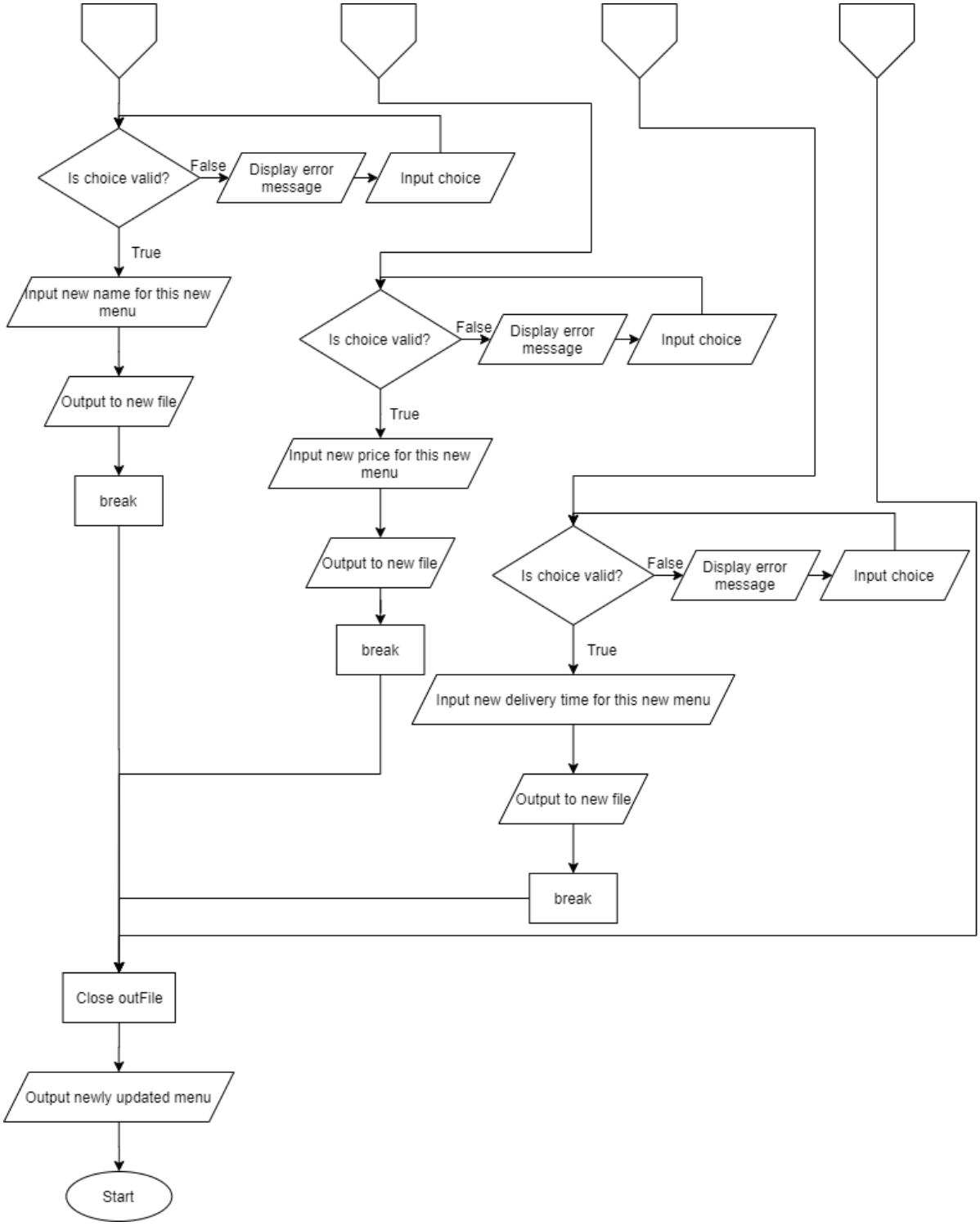


Flow chart for showMenu() function:

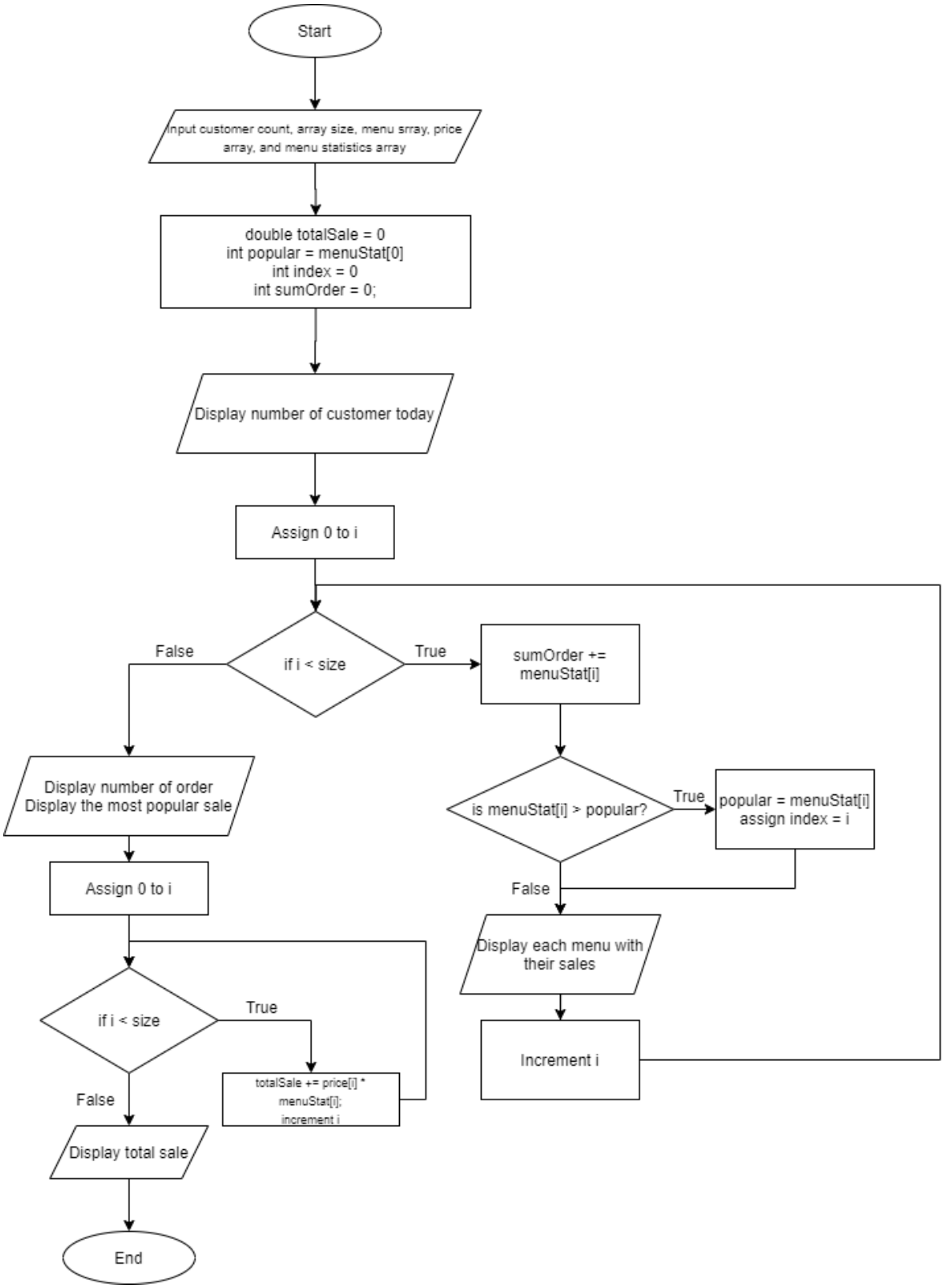


Flow chart of updateMenu() function:

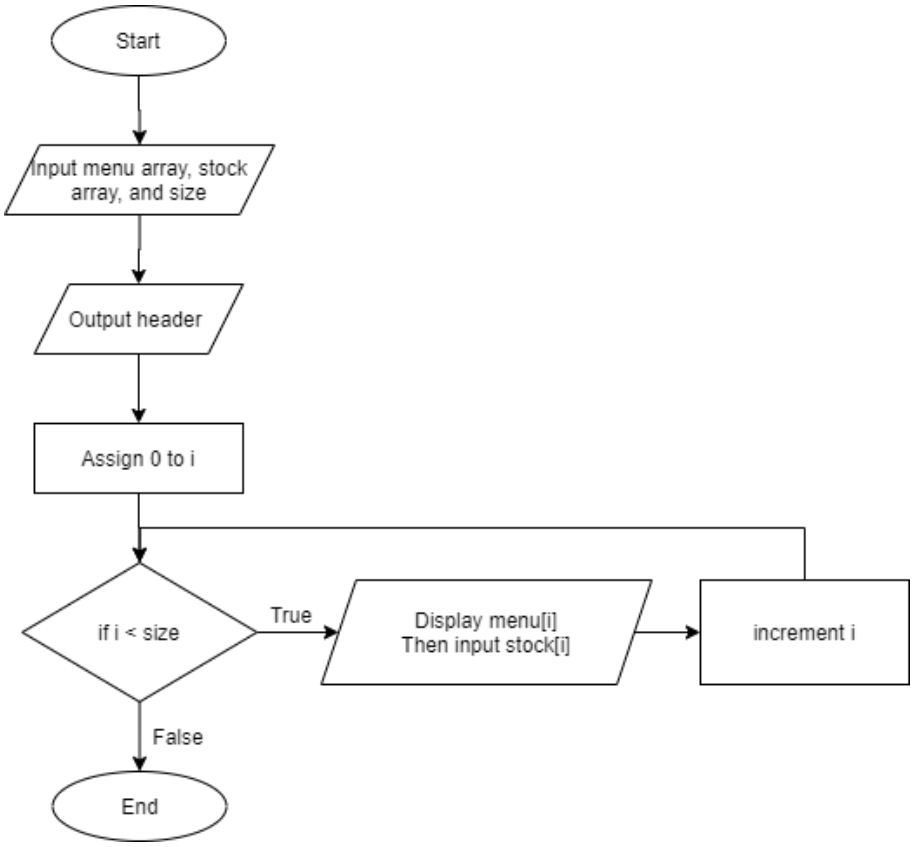



Is choice valid?
False
Display error message
Input choice
True
Input new name for this new menu
Output to new file
break
Is choice valid?
False
Display error message
Input choice
True
Input new price for this new menu
Output to new file
break
Is choice valid?
False
Display error message
Input choice
True
Input new delivery time for this new menu
Output to new file
break
Close outFile
Output newly updated menu
Start

Flow chart for showStat() function:



Flow chart for inputStock() function:



Program listings:

Two files are needed to run this program properly.

Source code: assignment2.cpp

Text file: menu.txt

Source code file:

```
/*
        Name                    : Ritchie Poh
        Matric Number           : 153765
        Email                   : ritchiepoh@student.usm.my
        Group                   : B
*/

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>

using namespace std;

//Function prototypes
int fileLines(ifstream&);
void showMenu(string*, double*, double*, int);
void updateMenu(string*, double*, double*, int);
void showStat(int, int, string*, double*, int*);
void inputStocks(string*, int*, int);

int main()
{
        ifstream inFile;
        inFile.open("menu.txt");
```

```
//Check for file error
if (!inFile)
{
	cout << "Error opening file" << endl;
}
else
{
	//Create dynamic arrays
	int arrSize = fileLines(inFile);
	int* stocks = new int[arrSize];
	int* menuStat = new int[arrSize];
	string* menu = new string[arrSize];
	double* price = new double[arrSize];
	double* delivery = new double[arrSize];

	//This for loop is to assign the statistics to zero first every time we run the program
	from the start.
	//And also input the file into parallel arrays
	for (int i = 0; i < arrSize; i++)
	{
		menuStat[i] = 0;
		getline(inFile, menu[i], '\t');
		inFile >> price[i];
		inFile.ignore(); //Ignore the tab between price and delivery time
		inFile >> delivery[i];
		inFile.ignore(); //Ignore \n character at end of each line
	}

	//Variable declaration
	int userChoice, guestCount = 0, order;
```

```
//To check if stock is entered every time the program is runned
bool stockFlag = false;


//Prompt the user to input to see is it customer or manager
//Or exit the program
cout << "\nHello, welcome to myBurger Lab!" << endl;
cout << "Press 0 if you are an admin of the restaurant."
        << "\nPress 1 if you are a customer."
        << "\nPress 2 to exit the program" << endl;
cin >> userChoice;

//Validate user input
while (userChoice < 0 || userChoice > 2)
{
        cout << "Invalid input, try again." << endl;
        cin >> userChoice;
}

//Sentinel value to exit a loop
while (userChoice != 2)
{
        //declare variable in here, so the variables will always start from a value I
want
        //Even after the menu is updated
        double orderSum = 0, deliverySum = 0;
        bool flag = false;
        ifstream File("menu.txt");
        int size = fileLines(File);

        if (userChoice == 1 && stockFlag)
```

```
{
        //Customer part
        showMenu(menu, price, delivery, size);
        cout << "\nChoose the number of the burger you would like to purchase."
                << "\nPress 0 if you want to exit." << endl;
        cin >> order;

        //Validate input
        while (order < 0 || order > size)
        {
                cout << "Sorry, we don't have this on our menu"
                        << "\nTry again." << endl;
                cin >> order;
        }

        //Sentinel value to exit the ordering loop
        while (order != 0)
        {
                //Check if there's still stock
                if (stocks[order - 1] > 0)
                {
                        //Show the order
                        cout << "\nYour order is " << menu[order - 1]
                                << "\nAnd the price is RM " << price[order - 1] << endl;

                        //Update the statistics
                        orderSum += price[order - 1];
                        deliverySum += delivery[order - 1];
                        menuStat[order - 1]++;
                        stocks[order - 1]--;
                        flag = true;
```

```
                    }
                    else
                    {
                        cout << "\nSorry this menu is sold out." << endl;
                    }

                    cout << "\nAnything else?" << endl;
                    cin >> order;
                }

                cout << "-----------------------------------------------------------------------------"
<< endl;
                cout << "Your total price is: RM " << orderSum << endl;
                cout << "Your delivery time will be approximately: " << deliverySum
<< "mins" << endl;
                cout << "-----------------------------------------------------------------------------"
<< endl;

                //If the user made order only make payment
                //And increment the customer count after payment made
                if (flag)
                {
                    //Make payment
                    double userPayment;
                    cout << "\nPlease make your payment. By entering the
amount that needs to be paid." << endl;
                    cin >> userPayment;

                    while (userPayment < orderSum)
                    {
                        cout << "\nPayment insufficient." << endl;
                        cin >> userPayment;
```

```
                }

                if (userPayment > orderSum)
                {
                        cout << "\nBalance is RM " << (userPayment -
orderSum) << endl;
                }
                guestCount++;
        }
        cout << "\nThank you for coming." << endl;
    }
    else
    {
        //Manager part
        //At the start of the program
        if (!stockFlag)
        {
                cout << "\nInput stock first." << endl;
                inputStocks(menu, stocks, size);
                stockFlag = true;
        }
        else
        {
                int managerChoice;
                cout << "\nPress 1 to update menu."
                        << "\nPress 2 to see statistics."
                        << "\nPress 3 to exit manager interface" << endl;
                cin >> managerChoice;

                //Validate user input
                while (managerChoice < 1 || managerChoice > 3)
```

```
{
        cout << "Invalid selection try again." << endl;
        cin >> managerChoice;
}

switch (managerChoice)
{
case 1:
{
        updateMenu(menu, price, delivery, size);
        break;
}
case 2:
{
        showStat(guestCount, arrSize, menu, price,
menuStat);

        break;
}
case 3:
{
        cout << "Exiting interface..." << endl;
        break;
}
default:
        cout << "Oops something went wrong, try again.\n"
<< endl;

        break;
}
    }
}

cout << "\nHello, welcome to myBurger Lab!" << endl;
```

```
                cout << "Press 0 if you are an admin of the restaurant."
                        << "\nPress 1 if you are a customer."
                        << "\nPress 2 to exit the program" << endl;
                cin >> userChoice;
                //Validate user input
                while (userChoice < 0 || userChoice > 2)
                {
                        cout << "Invalid input, try again." << endl;
                        cin >> userChoice;
                }
        }

        cout << "\nDone for today! We will be back tomorrow!" << endl;
        delete[] stocks;
        delete[] menu;
        delete[] price;
        delete[] delivery;
        delete[] menuStat;
    }

    inFile.close();
    return 0;
}

//Calculate the number of lines in a file
int fileLines(ifstream& inFile)
{
    int count = 0;
    string line;

    //Move the file to the beginning
```

```
	//So I can count all the lines
	inFile.clear();
	inFile.seekg(0L, ios::beg);
	while (!inFile.eof())
	{
		getline(inFile, line);
		count++;
	}
	//Move the file to the beginning
	//So the file is always from the start as a standard
	inFile.clear();
	inFile.seekg(0L, ios::beg);
	return count;
}

void showMenu(string* menu, double* price, double* delivery, int size)
{
	int count = 1;

	cout << "\n\n\t\tMenu for myBurger Lab\n\n" << endl;
	cout << setw(4) << left << "Num"
		<< setw(20) << left << "Menu" << "\t\t\t"
		<< setw(8) << left << "Price" << "\t"
		<< "Delivery Time" << endl;
	cout << "-------------------------------------------------------------------------" << endl;

	for (int i = 0; i < size; i++)
	{
		cout << setw(4) << left << count++
			<< setw(20) << left << menu[i] << "\t\t\t"
			<< "RM " << setprecision(2) << fixed << price[i] << "\t"
```

```
                << delivery[i] << "mins" << endl;

        }
        cout << "-----------------------------------------------------------------------------" << endl;
}

void updateMenu(string* menu, double* price, double* delivery, int size)
{
        int userChoice;

        cout << "This is the current menu." << endl;
        showMenu(menu, price, delivery, size);
        cout << "\nPress 1 if you want to update a menu."
                << "\nPress 2 if you want to update a price."
                << "\nPress 3 if you want to update a delivery time" << endl;
        cin >> userChoice;

        //Validate user input
        while (userChoice < 1 || userChoice > 3)
        {
                cout << "Invalid selection, try again." << endl;
                cin >> userChoice;
        }

        //Overwrite the existing file
        fstream outFile("menu.txt", ios::out);

        switch (userChoice)
        {
        case 1:
        {
```

```
int choice;
string newMenu;
cout << "Which menu do you want to update?"
        << "\nEnter the number of the menu." << endl;
cin >> choice;



//Validate user input
while (choice < 1 || choice > size)
{
        cout << "Invalid selection, try again." << endl;
        cin >> choice;
}

cout << "Enter the new name for this menu." << endl;
cin.ignore();
getline(cin, menu[choice - 1]);

outFile << menu[0] << "\t" << setprecision(2) << fixed << price[0] << "\t"
        << delivery[0];
for (int i = 1; i < size; i++)
{
        outFile << "\n" << menu[i] << "\t" << setprecision(2) << fixed << price[i] <<
"\t"
                << delivery[i];
}

break;
}

case 2:
```

```
{
        int choice;
        cout << "Which menu do you want to update?"
                << "\nEnter the number of the menu." << endl;
        cin >> choice;

        //Validate user input
        while (choice < 1 || choice > size)
        {
                cout << "Invalid selection, try again." << endl;
                cin >> choice;
        }

        cout << "Enter new price in \"RM\"" << endl;
        cin >> price[choice - 1];

        outFile << menu[0] << "\t" << setprecision(2) << fixed << price[0] << "\t"
                << delivery[0];
        for (int i = 1; i < size; i++)
        {
                outFile << "\n" << menu[i] << "\t" << setprecision(2) << fixed << price[i] <<
"\t"
                        << delivery[i];
        }

        break;
}

case 3:
{
        int choice;
```

```
		cout << "Which menu do you want to update?"
			<< "\nEnter the number of the menu." << endl;
		cin >> choice;

		//Validate user input
		while (choice < 1 || choice > size)
		{
			cout << "Invalid selection, try again." << endl;
			cin >> choice;
		}

		cout << "Enter new time in \"mins\"" << endl;
		cin >> delivery[choice - 1];

		outFile << menu[0] << "\t" << setprecision(2) << fixed << price[0] << "\t"
			<< delivery[0];
		for (int i = 1; i < size; i++)
		{
			outFile << "\n" << menu[i] << "\t" << setprecision(2) << fixed << price[i] << "\t"
				<< delivery[i];
		}

		break;
	}

	default:
	{
		cout << "Selection error." << endl;
		break;
	}
```

```
	}

	outFile.close();

	cout << "\nNewly updated menu looks like this." << endl;
	showMenu(menu, price, delivery, size);
}

//Show the statistics of the restaurant
void showStat(int customer, int arrSize, string* menu, double* price, int* menuStat)
{
	double totalSale = 0;
	int popular = menuStat[0], index = 0, sumOrder = 0;
	cout << "\n--------------------------------------------------------------------------------" << endl;
	cout << "Number of customer for today: "
		<< customer << endl;

	//Print the sale of each burger and determine the most popular sale
	//Also determine the number of orders
	cout << "This is the sale of every burger:\n" << endl;
	for (int i = 0; i < arrSize; i++)
	{
		sumOrder += menuStat[i]; //Calculate the number of orders
		if (menuStat[i] > popular)
		{
			popular = menuStat[i];
			index = i;
		}
		cout << setw(20) << left << menu[i] << ": " << menuStat[i] << endl;
	}
```

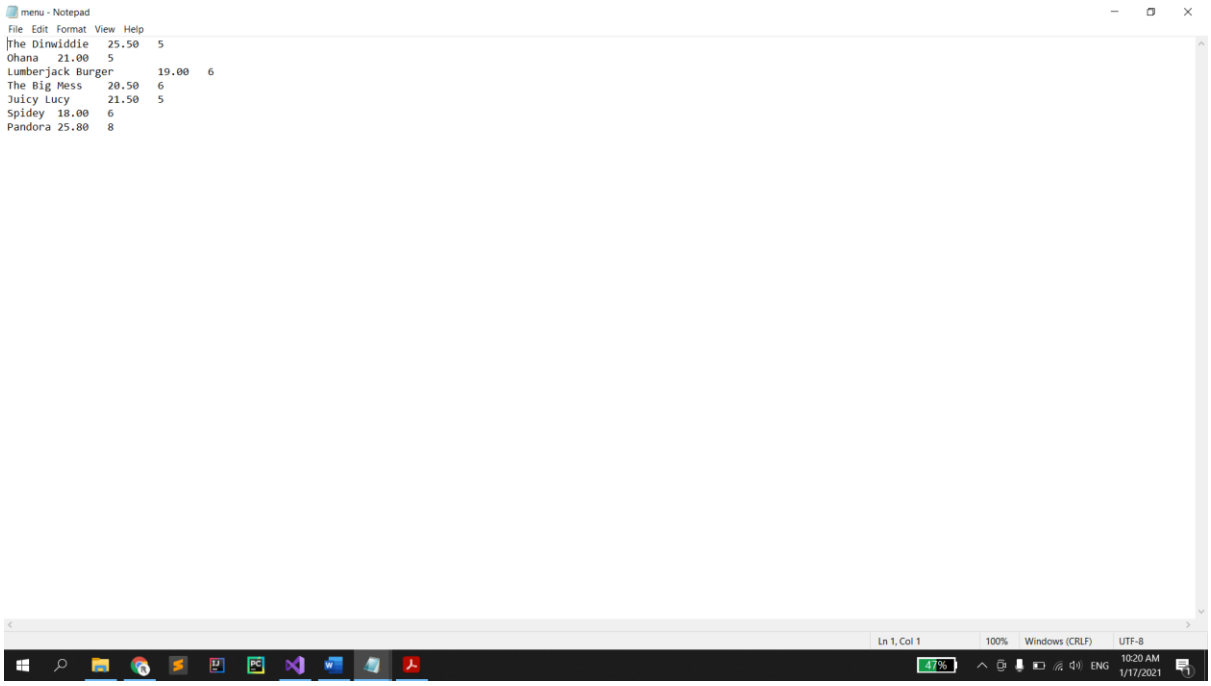
```
	cout << "\nThe number of orders by customer today: " << sumOrder << endl;
	cout << "The most popular sale for today is " << menu[index]
		<< " with " << popular << " number of sales." << endl;

	for (int i = 0; i < arrSize; i++)
	{
		totalSale += price[i] * menuStat[i];
	}
	cout << "\nTotal sale so far today is RM " << setprecision(2) << fixed << totalSale << endl;
	cout << "\n--------------------------------------------------------------------------------" << endl;
}

//Input the number of stocks for each menu
void inputStocks(string* menu, int* stocks, int size)
{
	cout << "\nInput the stocks of every burger in the menu." << endl;
	for (int i = 0; i < size; i++)
	{
		cout << setw(20) << left << menu[i] << ": ";
		cin >> stocks[i];
	}
}
```

<u>menu.txt file:</u>

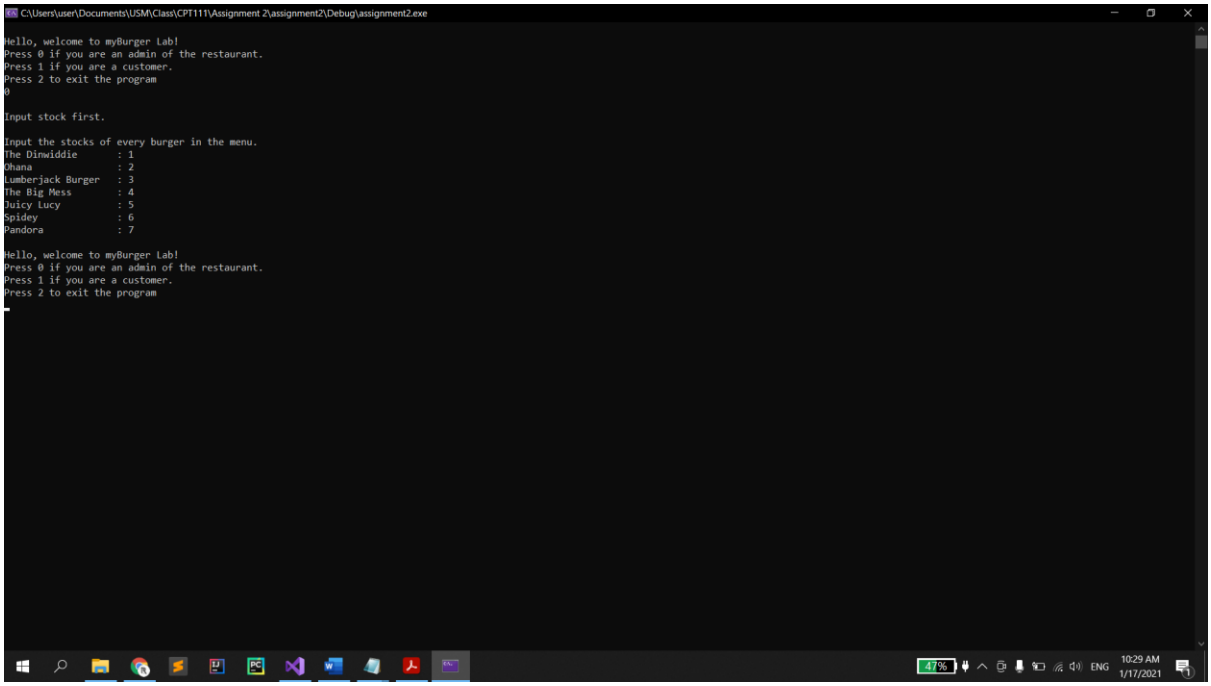
```
The Dinwiddie	25.50	5
Ohana	21.00	5
Lumberjack Burger	19.00	6
The Big Mess	20.50	6
Juicy Lucy	21.50	5
Spidey	18.00	6
Pandora	25.80	8
```

Screenshots of sample IO files

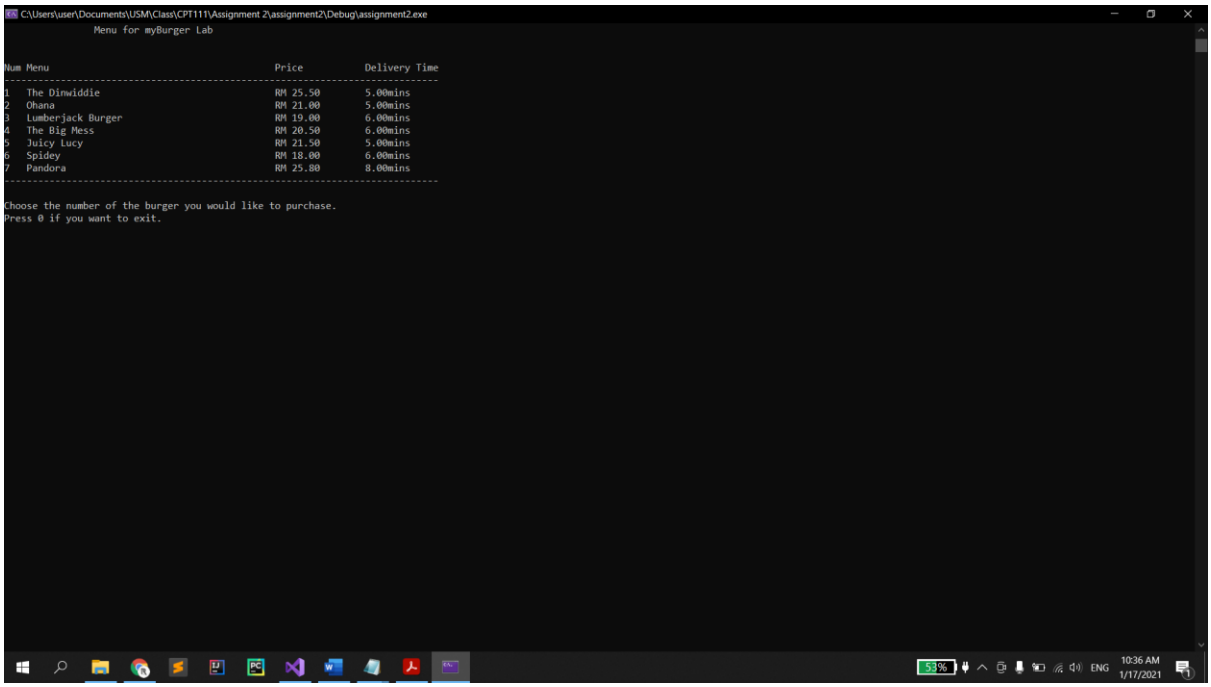
menu.txt



Interface to input stocks

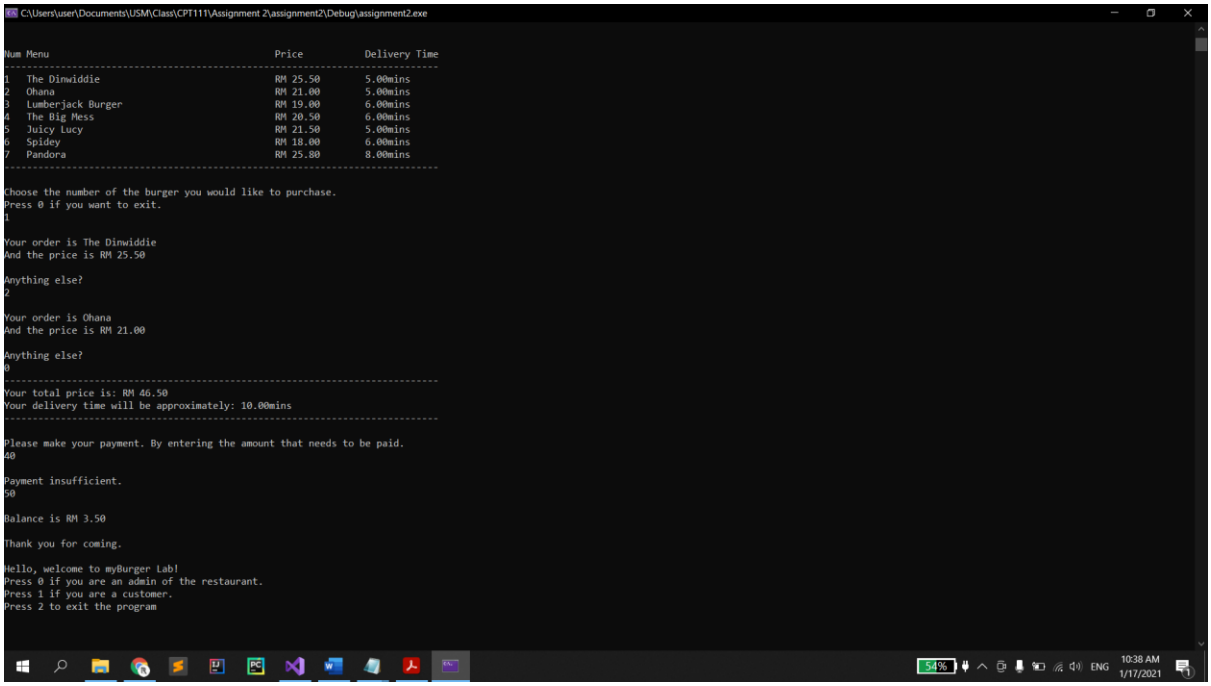


The menu interface



```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe
                Menu for myBurger Lab

Num Menu                                        Price          Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50       5.00mins
2   Ohana                                       RM 21.00       5.00mins
3   Lumberjack Burger                           RM 19.00       6.00mins
4   The Big Mess                                RM 20.50       6.00mins
5   Juicy Lucy                                  RM 21.50       5.00mins
6   Spidey                                      RM 18.00       6.00mins
7   Pandora                                     RM 25.80       8.00mins
-----------------------------------------------------------------------------

Choose the number of the burger you would like to purchase.
Press 0 if you want to exit.
```

Interface for customer making order and making payment



```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe

Num Menu                                        Price          Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50       5.00mins
2   Ohana                                       RM 21.00       5.00mins
3   Lumberjack Burger                           RM 19.00       6.00mins
4   The Big Mess                                RM 20.50       6.00mins
5   Juicy Lucy                                  RM 21.50       5.00mins
6   Spidey                                      RM 18.00       6.00mins
7   Pandora                                     RM 25.80       8.00mins
-----------------------------------------------------------------------------

Choose the number of the burger you would like to purchase.
Press 0 if you want to exit.
1

Your order is The Dinwiddie
And the price is RM 25.50

Anything else?
2

Your order is Ohana
And the price is RM 21.00

Anything else?
0
-----------------------------------------------------------------------------
Your total price is: RM 46.50
Your delivery time will be approximately: 10.00mins
-----------------------------------------------------------------------------

Please make your payment. By entering the amount that needs to be paid.
40

Payment insufficient.
50

Balance is RM 3.50

Thank you for coming.

Hello, welcome to myBurger Lab!
Press 0 if you are an admin of the restaurant.
Press 1 if you are a customer.
Press 2 to exit the program
```

Manager interface

```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe

Press 1 to update menu.
Press 2 to see statistics.
Press 3 to exit manager interface
```

Update menu interface

```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe
Press 1 to update menu.
Press 2 to see statistics.
Press 3 to exit manager interface
1
This is the current menu.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        5.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 18.00        6.00mins
7   Pandora                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Press 1 if you want to update a menu.
Press 2 if you want to update a price.
Press 3 if you want to update a delivery time
1
Which menu do you want to update?
Enter the number of the menu.
7
Enter the new name for this menu.
Whopper

Newly updated menu looks like this.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        5.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 18.00        6.00mins
7   Whopper                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Hello, welcome to myBurger Lab!
Press 0 if you are an admin of the restaurant.
```

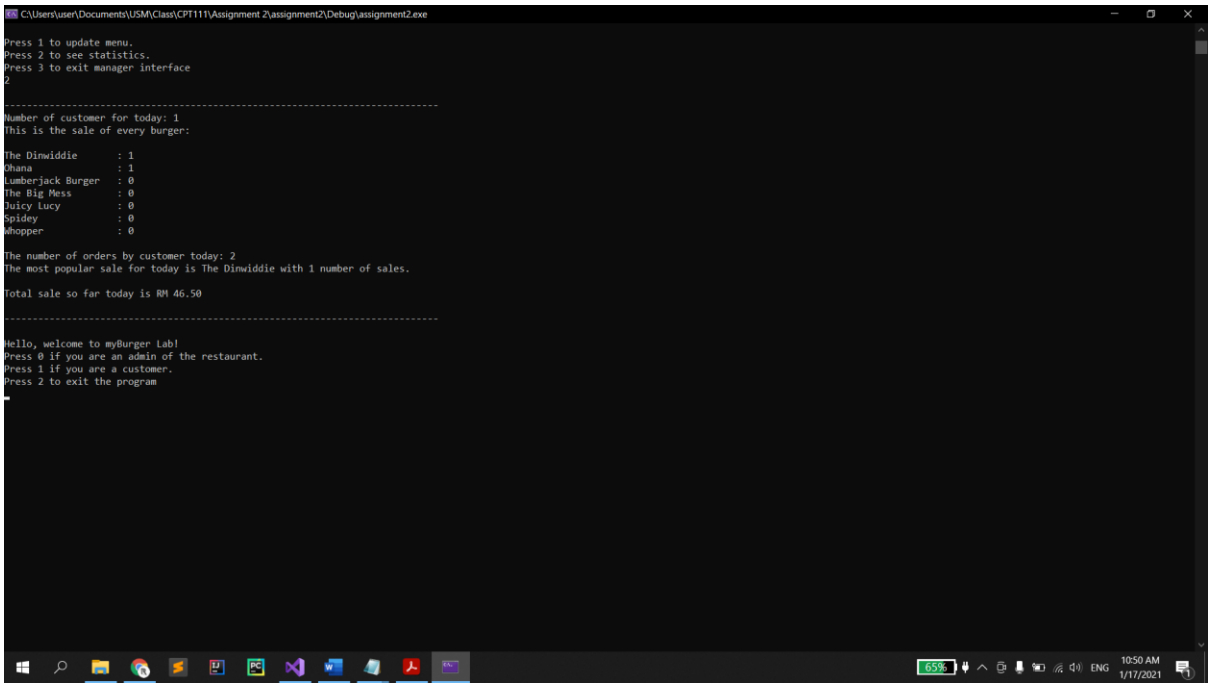
Update price interface

```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe
Press 1 to update menu.
Press 2 to see statistics.
Press 3 to exit manager interface
1
This is the current menu.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        5.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 18.00        6.00mins
7   Whopper                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Press 1 if you want to update a menu.
Press 2 if you want to update a price.
Press 3 if you want to update a delivery time
2
Which menu do you want to update?
Enter the number of the menu.
6
Enter new price in "RM"
19.5

Newly updated menu looks like this.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        5.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 19.50        6.00mins
7   Whopper                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Hello, welcome to myBurger Lab!
Press 0 if you are an admin of the restaurant.
```

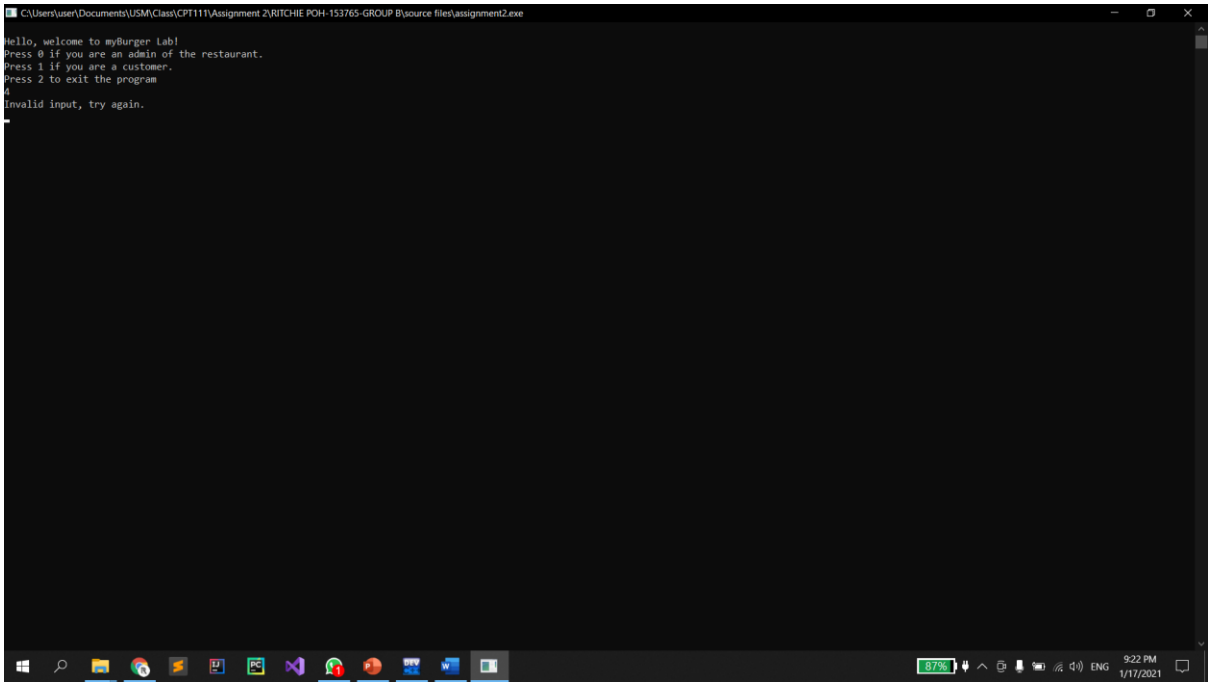
Update delivery time interface

```
C:\Users\user\Documents\USM\Class\CPT111\Assignment 2\assignment2\Debug\assignment2.exe
Press 1 to update menu.
Press 2 to see statistics.
Press 3 to exit manager interface
1
This is the current menu.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        5.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 19.50        6.00mins
7   Whopper                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Press 1 if you want to update a menu.
Press 2 if you want to update a price.
Press 3 if you want to update a delivery time
1
Which menu do you want to update?
Enter the number of the menu.
1
Enter new time in "mins"
6

Newly updated menu looks like this.

                Menu for myBurger Lab

Num Menu                                        Price           Delivery Time
-----------------------------------------------------------------------------
1   The Dinwiddie                               RM 25.50        6.00mins
2   Ohana                                       RM 21.00        5.00mins
3   Lumberjack Burger                           RM 19.00        6.00mins
4   The Big Mess                                RM 20.50        6.00mins
5   Juicy Lucy                                  RM 21.50        5.00mins
6   Spidey                                      RM 19.50        6.00mins
7   Whopper                                     RM 25.80        8.00mins
-----------------------------------------------------------------------------

Hello, welcome to myBurger Lab!
Press 0 if you are an admin of the restaurant.
```

See statistics interface



Validation interface



This validation interface is same across the program.

This code is also uploaded onto Git Hub for storage.

https://github.com/RitchieP/CPT111assignment2