UNIVERSITI TEKNOLOGI MARA

PERAK BRANCH, TAPAH CAMPUS

35400 Tapah Road

College of Computing, Informatics and Media

Fundamentals of Algorithms & Computer Problem Solving

(CSC126)

Group Project Food Ordering System

(Assessment 4)

NO.	NAMA AHLI KUMPULAN	NO. PELAJAR
1.	SHEIKH ADAM BAJUNID BIN MOHD FAISAL	2023135385
2.	MOHAMAD IMAN MUZAKKIR BIN ISMAIL	2023159911
3.	MUHAMAD AZIM HAFIZI BIN CHE MAT	2023172751
4.	AMMAR BIN AHMAD MUDZFIR	2023103981

LECTURER:
DR. MOHD. FAAIZIE BIN DARMAWAN

Proposed Project Summary:
The Food Ordering System is a C++ application designed to facilitate the process of ordering food online. The system aims to provide a convenient and user-friendly platform for customers to browse the restaurant menu, place orders, and make payments without any human interaction.

Here are the 5 key features of the program we made:

1. Menu Display
    - The system displays a menu where users can select food items by entering the corresponding item codes. The menu includes options such as pizza, burger, ice cream, and sandwich.
2. Quantity and Addon Management
    - Users can specify the quantity of each food item they want to order. Additionally, the system prompts users if they would like to add any addons to their food items.
3. Cost and Government Tax Calculations
    - The system calculates the individual cost for each food item, as well as the total cost of all ordered items. It tracks the quantity of each food item and maintains a running total using arrays. The system also applies a 6% government tax to the total cost of the ordered food.
4. The ability to select Payment Methods
    - Users are presented with payment method options and can choose one that suits them best. The system handles the payment amount and calculates the change, if it is required.
5. Order Receipt Generation
    - Upon completion of the order, the system generates an order receipt that includes details such as the total cost of the food items, government tax, payment method, and change (if any). The receipt provides a summary of the order for the user's reference or for future tax reasons.

Objective of the project

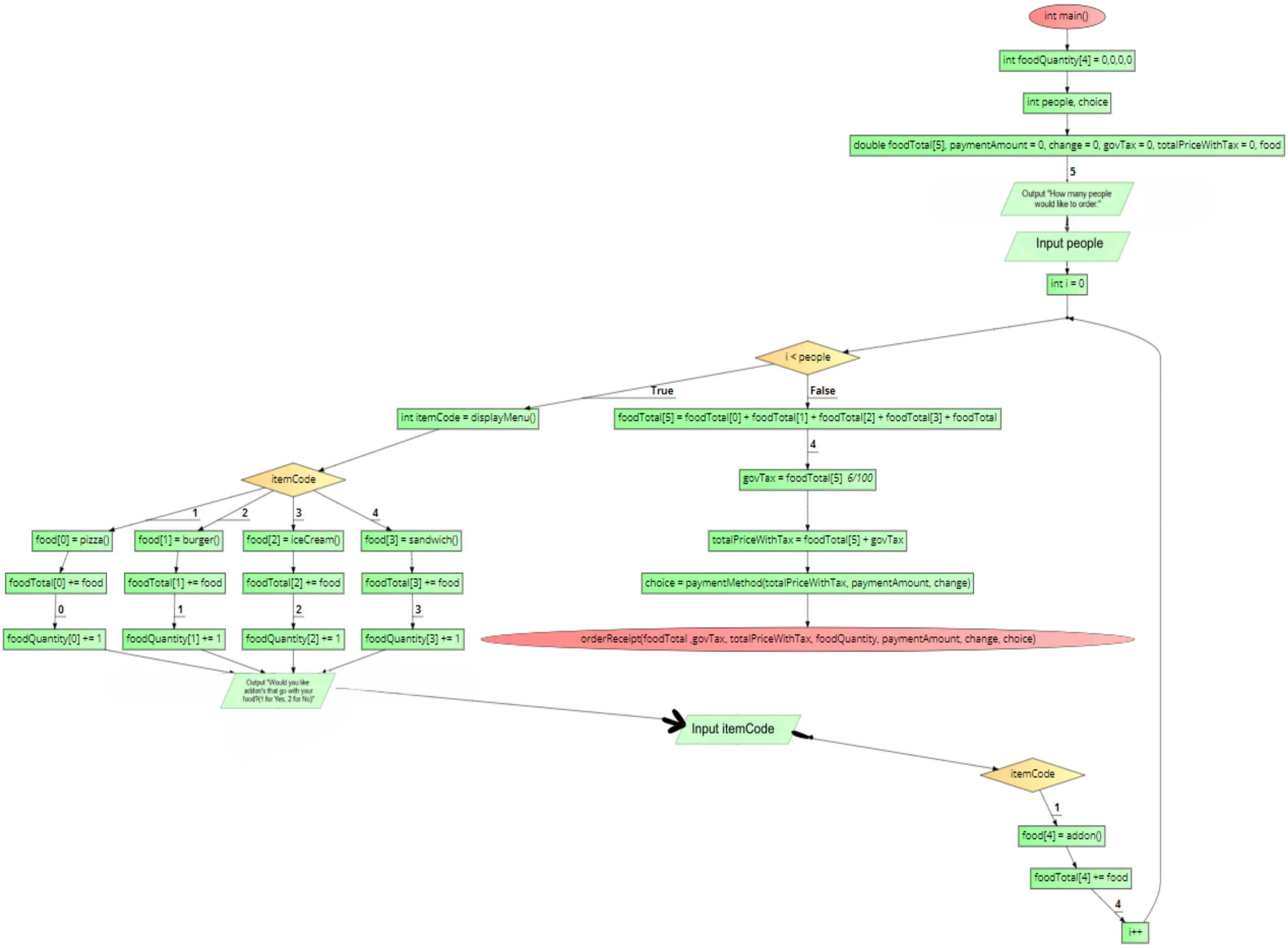
The purpose for this C++ program is to help users to make their preference order very fast and easy. In our food menu program, there are a number of codes that the user must choose and enter based on their preferences. This program will also ask for payment methods such as through cash, credit or debit card, E-wallet and FPX. After that, it will display the receipt that will show foods, subtotal, 6 percent service tax, total price, payment method, payment amount and lastly change. Based on the program, the calculation is fixed, and the user cannot change any value by themself so if they want to make any changes then they will need to ask for changes through real communication.

Analysis – Input/Process/Output

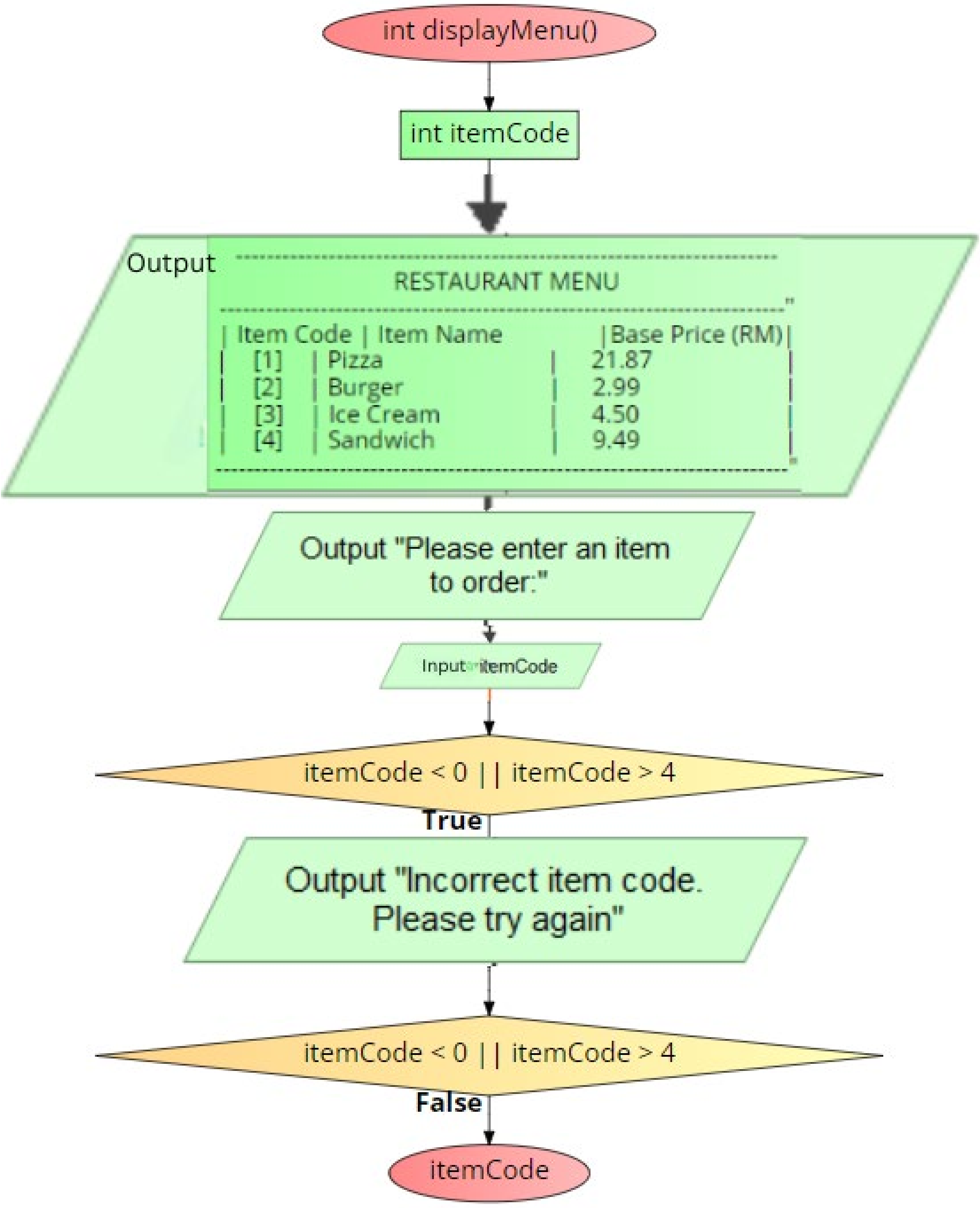
INPUT	people , itemcode, choice, addon, price, type
PROCESS	food[0] = foodFunctionName() ; foodTotal[0] += food[0]; foodQuantity[0] += 1;
OUTPUT	orderReceipt

Flowcharts:

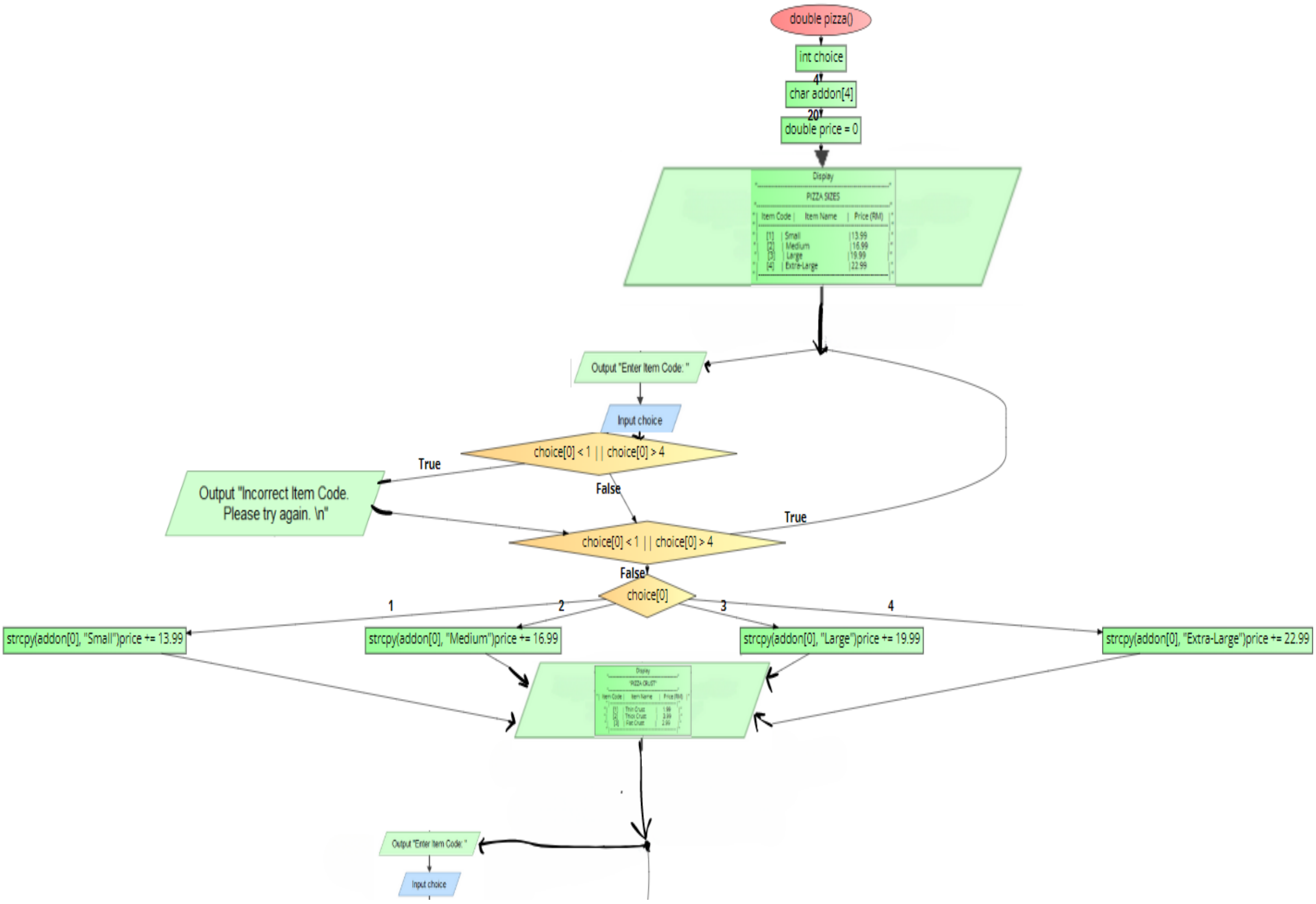
1. Main function


int main()
int foodQuantity[4] = 0,0,0,0
int people, choice
double foodTotal[5], paymentAmount = 0, change = 0, govTax = 0, totalPriceWithTax = 0, food
5
Output "How many people would like to order."
Input people
int i = 0
i < people
True
False
int itemCode = displayMenu()
foodTotal[5] = foodTotal[0] + foodTotal[1] + foodTotal[2] + foodTotal[3] + foodTotal
4
govTax = foodTotal[5] 6/100
totalPriceWithTax = foodTotal[5] + govTax
choice = paymentMethod(totalPriceWithTax, paymentAmount, change)
orderReceipt(foodTotal ,govTax, totalPriceWithTax, foodQuantity, paymentAmount, change, choice)
itemCode
1
2
3
4
food[0] = pizza()
food[1] = burger()
food[2] = iceCream()
food[3] = sandwich()
foodTotal[0] += food
foodTotal[1] += food
foodTotal[2] += food
foodTotal[3] += food
0
1
2
3
foodQuantity[0] += 1
foodQuantity[1] += 1
foodQuantity[2] += 1
foodQuantity[3] += 1
Output "Would you like addon's that go with your food?(1 for Yes, 2 for No)"
Input itemCode
itemCode
1
food[4] = addon()
foodTotal[4] += food
4
i++

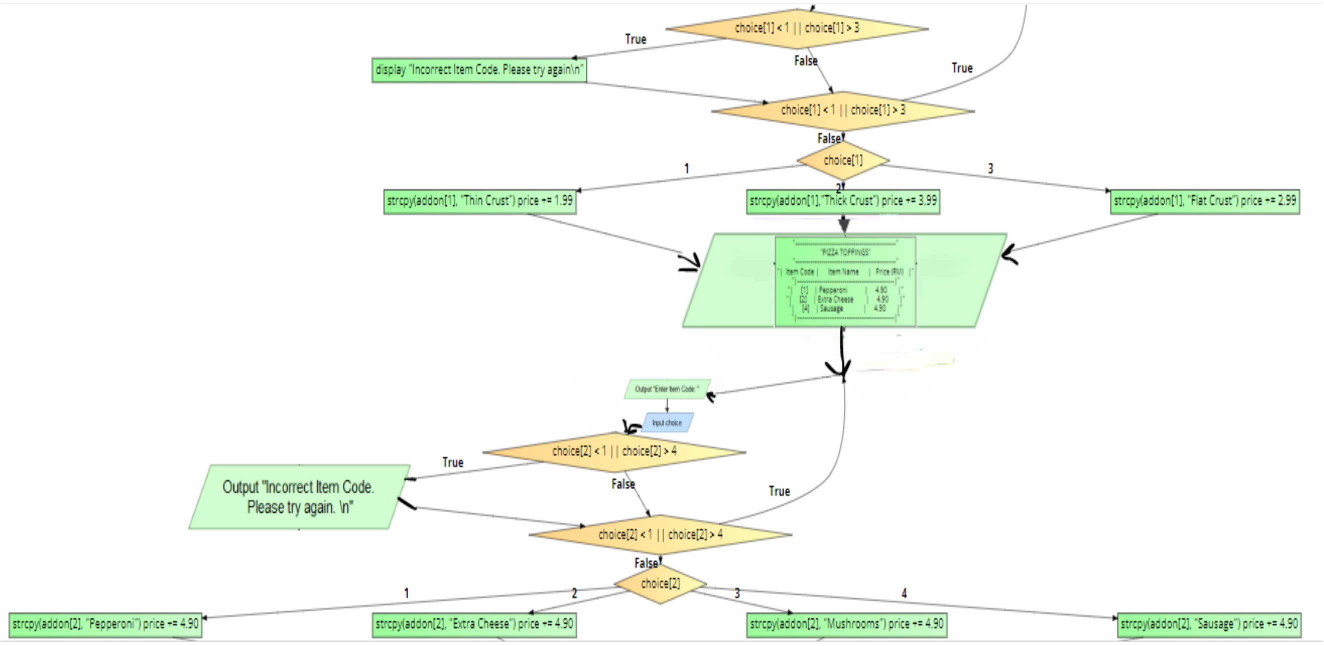
2. DisplayMenu Function


int displayMenu()
int itemCode
Output
RESTAURANT MENU
Item Code	Item Name	Base Price (RM)
[1]	Pizza	21.87
[2]	Burger	2.99
[3]	Ice Cream	4.50
[4]	Sandwich	9.49
Output "Please enter an item to order:"
Input itemCode
itemCode < 0 || itemCode > 4
True
Output "Incorrect item code. Please try again"
itemCode < 0 || itemCode > 4
False
itemCode

3. Pizza Function(1)

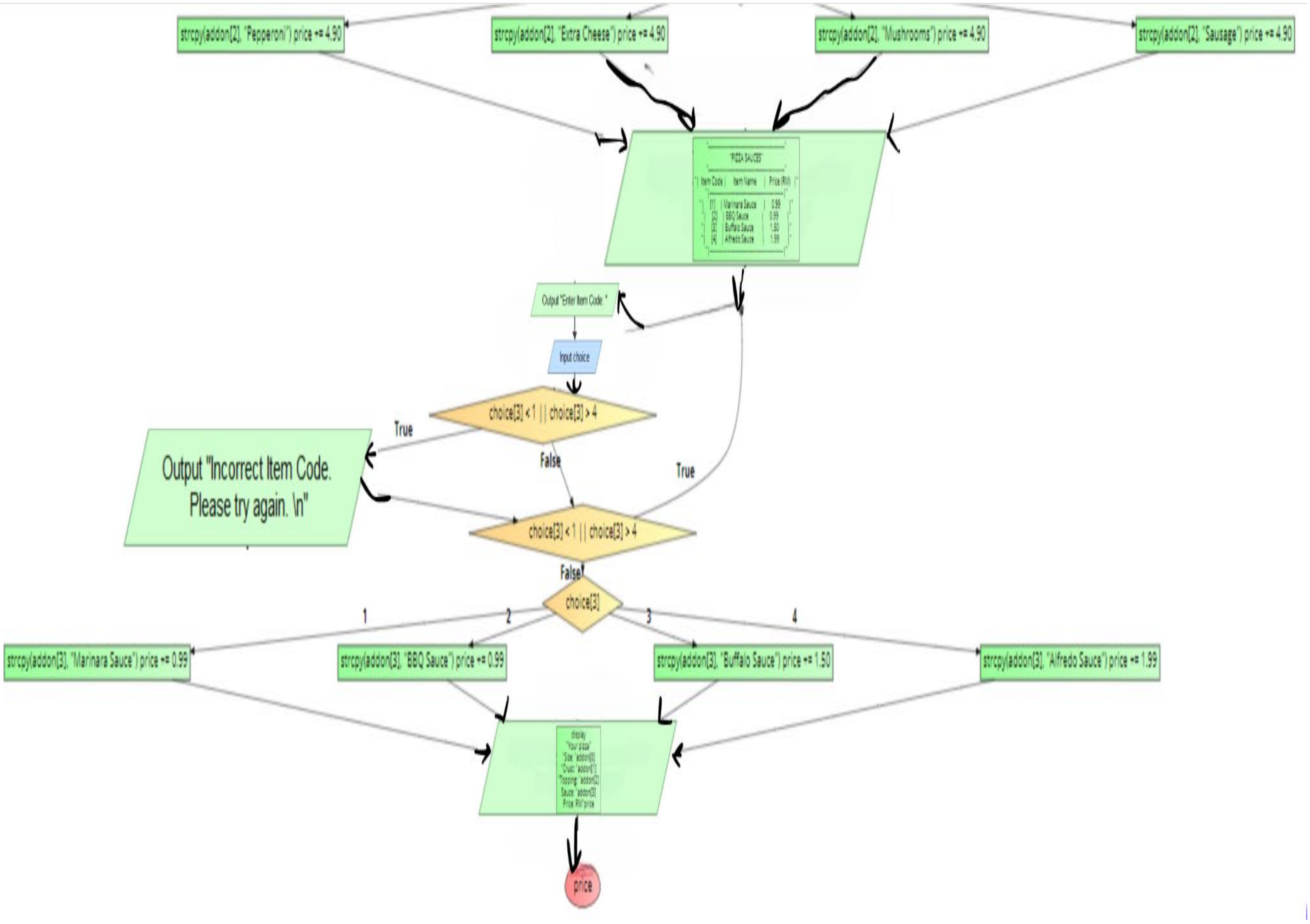

double pizza()
int choice
4
char addon[4]
20
double price = 0
Display
PIZZA SIZES
Item Code | Item Name | Price (RM)
[1] | Small | 13.99
[2] | Medium | 16.99
[3] | Large | 19.99
[4] | Extra-Large | 22.99
Output "Enter Item Code: "
Input choice
choice[0] < 1 || choice[0] > 4
True
Output "Incorrect Item Code. Please try again. \n"
False
choice[0] < 1 || choice[0] > 4
True
False
choice[0]
1
2
3
4
strcpy(addon[0], "Small")price += 13.99
strcpy(addon[0], "Medium")price += 16.99
strcpy(addon[0], "Large")price += 19.99
strcpy(addon[0], "Extra-Large")price += 22.99
Display
"PIZZA CRUST"
Item Code | Item Name | Price (RM)
[1] | Thin Crust | 1.99
[2] | Thick Crust | 3.99
[3] | Flat Crust | 2.99
Output "Enter Item Code: "
Input choice

Pizza Function(2)

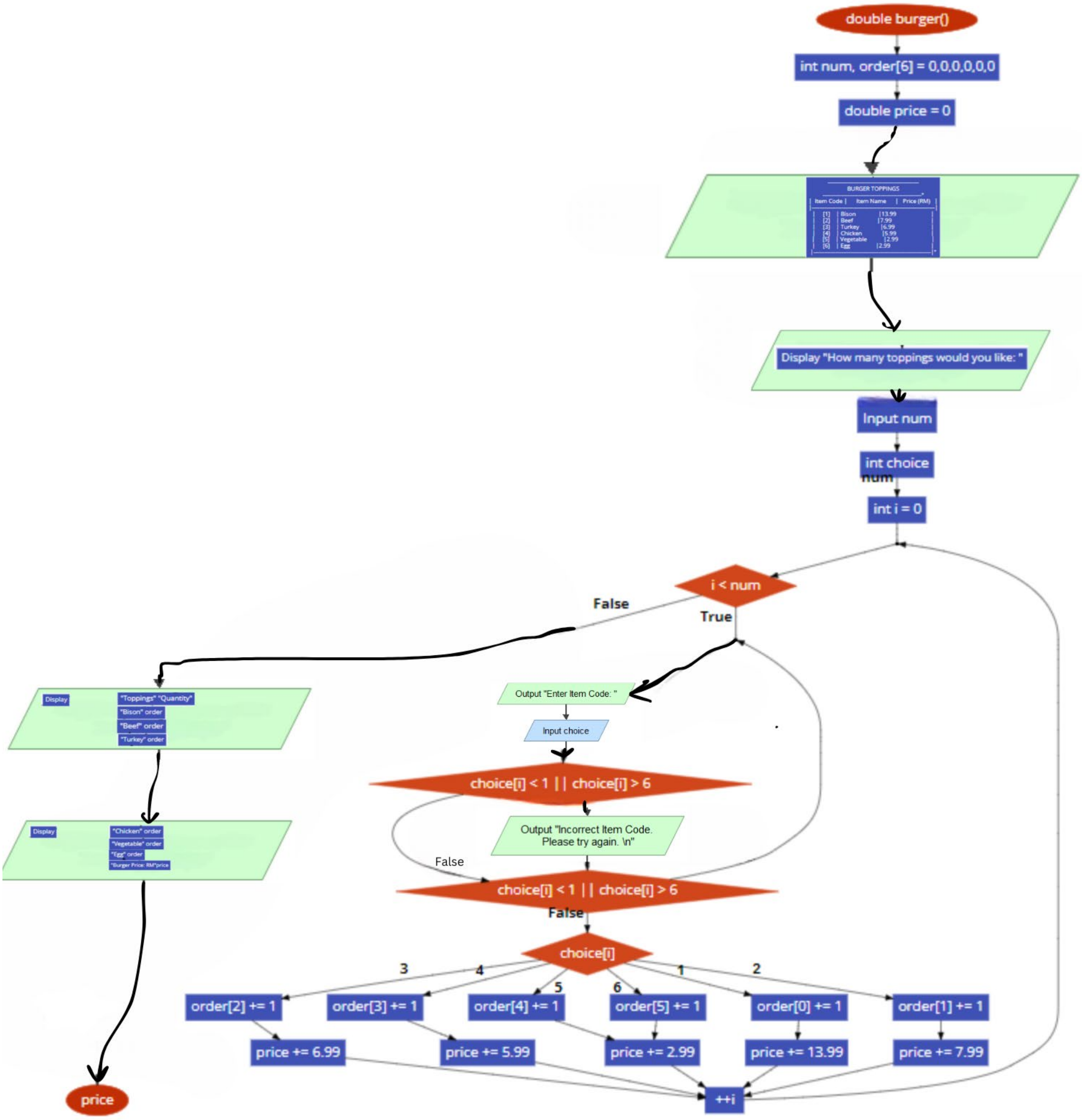


choice[1] < 1 || choice[1] > 3
True
display "Incorrect Item Code. Please try again\n"
False
True
choice[1] < 1 || choice[1] > 3
False
choice[1]
1
2
3
strcpy(addon[1], "Thin Crust") price += 1.99
strcpy(addon[1],"Thick Crust") price += 3.99
strcpy(addon[1], "Flat Crust") price += 2.99
"PIZZA TOPPINGS"
| Item Code | Item Name | Price (RM) |
[1] | Pepperoni | 4.90
[2] | Extra Cheese | 4.90
[4] | Sausage | 4.90
Output "Enter Item Code: "
Input choice
choice[2] < 1 || choice[2] > 4
True
Output "Incorrect Item Code. Please try again. \n"
False
True
choice[2] < 1 || choice[2] > 4
False
choice[2]
1
2
3
4
strcpy(addon[2], "Pepperoni") price += 4.90
strcpy(addon[2], "Extra Cheese") price += 4.90
strcpy(addon[2], "Mushrooms") price += 4.90
strcpy(addon[2], "Sausage") price += 4.90

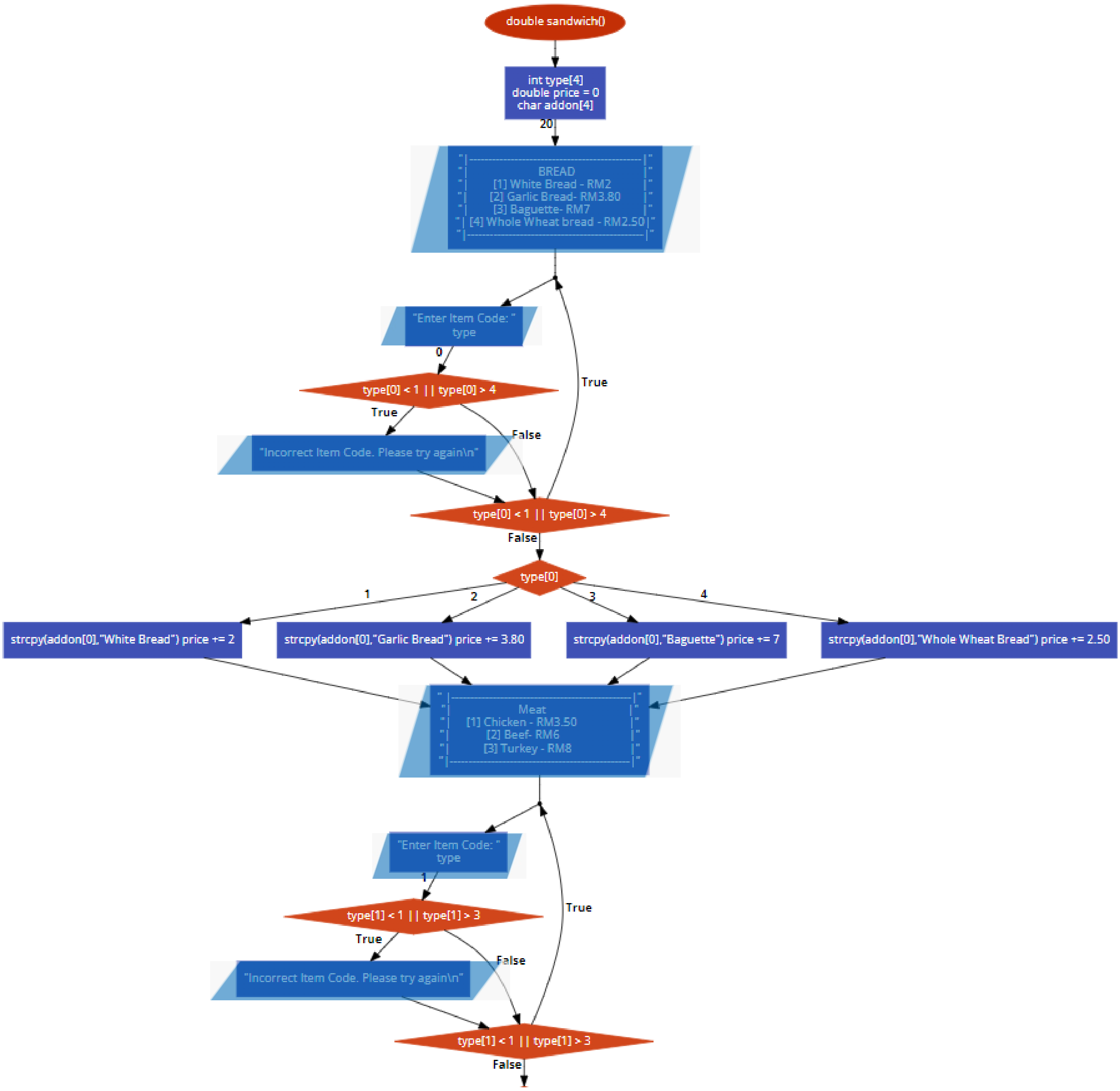
Pizza Function(3)


strcpy(addon[2], "Pepperoni") price += 4.90
strcpy(addon[2], "Extra Cheese") price += 4.90
strcpy(addon[2], "Mushrooms") price += 4.90
strcpy(addon[2], "Sausage") price += 4.90
"PIZZA SAUCES"
Item Code | Item Name | Price (RM)
[1] | Marinara Sauce | 0.99
[2] | BBQ Sauce | 0.99
[3] | Buffalo Sauce | 1.50
[4] | Alfredo Sauce | 1.99
Output "Enter Item Code: "
Input choice
choice[3] < 1 || choice[3] > 4
True
False
Output "Incorrect Item Code. Please try again. \n"
True
choice[3] < 1 || choice[3] > 4
False
choice[3]
1
2
3
4
strcpy(addon[3], "Marinara Sauce") price += 0.99
strcpy(addon[3], "BBQ Sauce") price += 0.99
strcpy(addon[3], "Buffalo Sauce") price += 1.50
strcpy(addon[3], "Alfredo Sauce") price += 1.99
display
"Your pizza"
"Size: "addon[0]
"Crust: "addon[1]
"Topping: "addon[2]
Sauce: "addon[3]
Price: RM"price
price

4. Burger Function

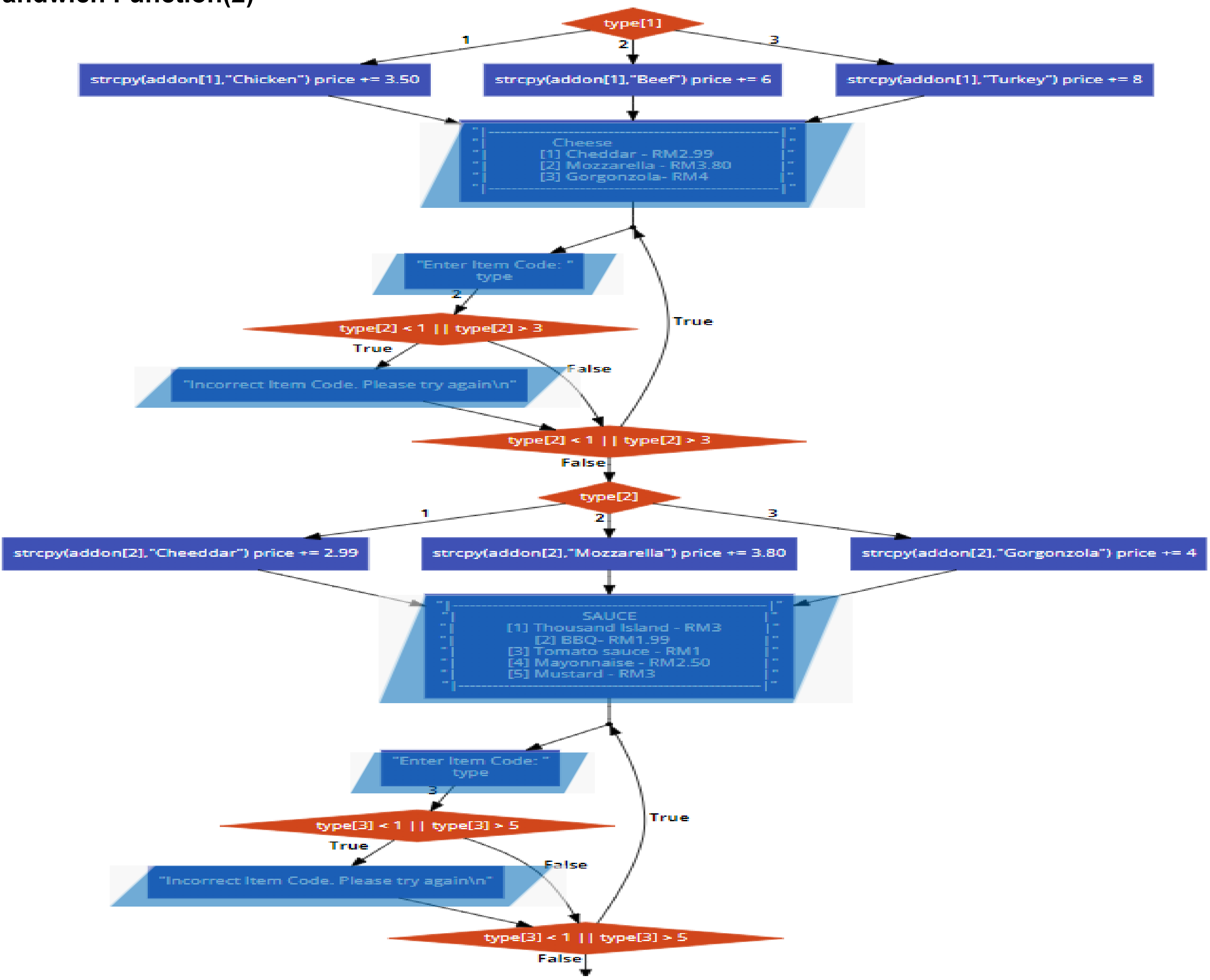


5. Sandwich Function(1)



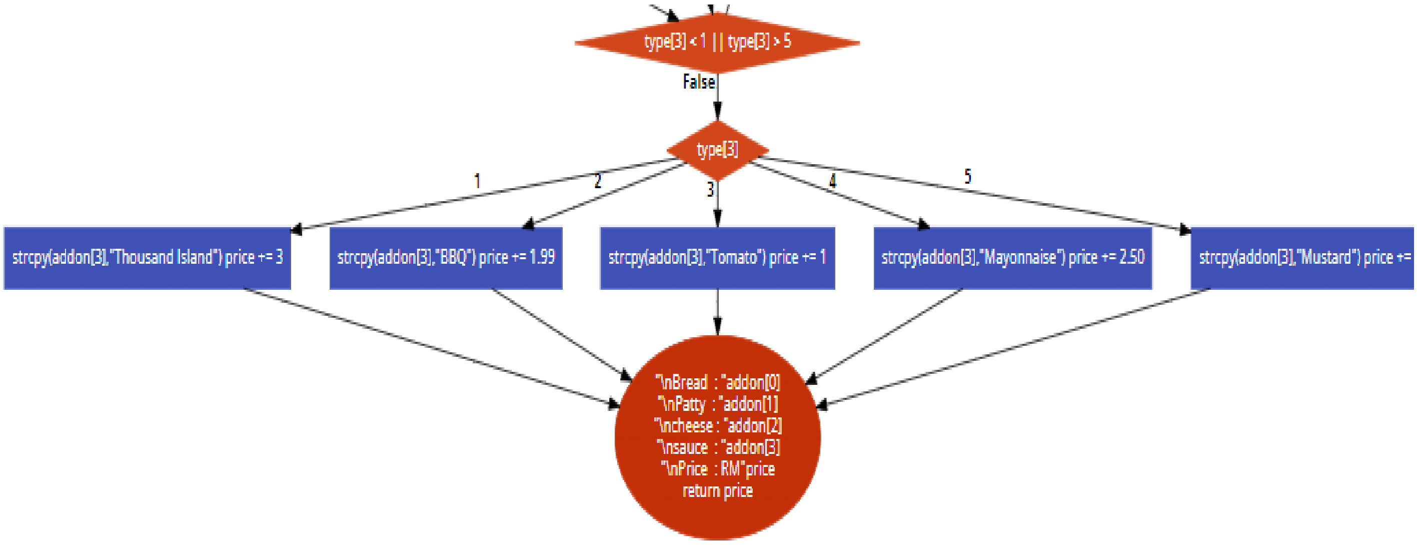
double sandwich()
int type[4]
double price = 0
char addon[4]
20
"|--|"
"| BREAD |"
"| [1] White Bread - RM2 |"
"| [2] Garlic Bread- RM3.80 |"
"| [3] Baguette- RM7 |"
"| [4] Whole Wheat bread - RM2.50 |"
"|--|"
"Enter Item Code: "
type
0
type[0] < 1 || type[0] > 4
True
"Incorrect Item Code. Please try again\n"
False
True
type[0] < 1 || type[0] > 4
False
type[0]
1
2
3
4
strcpy(addon[0],"White Bread") price += 2
strcpy(addon[0],"Garlic Bread") price += 3.80
strcpy(addon[0],"Baguette") price += 7
strcpy(addon[0],"Whole Wheat Bread") price += 2.50
"|--|"
"| Meat |"
"| [1] Chicken - RM3.50 |"
"| [2] Beef- RM6 |"
"| [3] Turkey - RM8 |"
"|--|"
"Enter Item Code: "
type
1
type[1] < 1 || type[1] > 3
True
"Incorrect Item Code. Please try again\n"
False
True
type[1] < 1 || type[1] > 3
False

Sandwich Function(2)

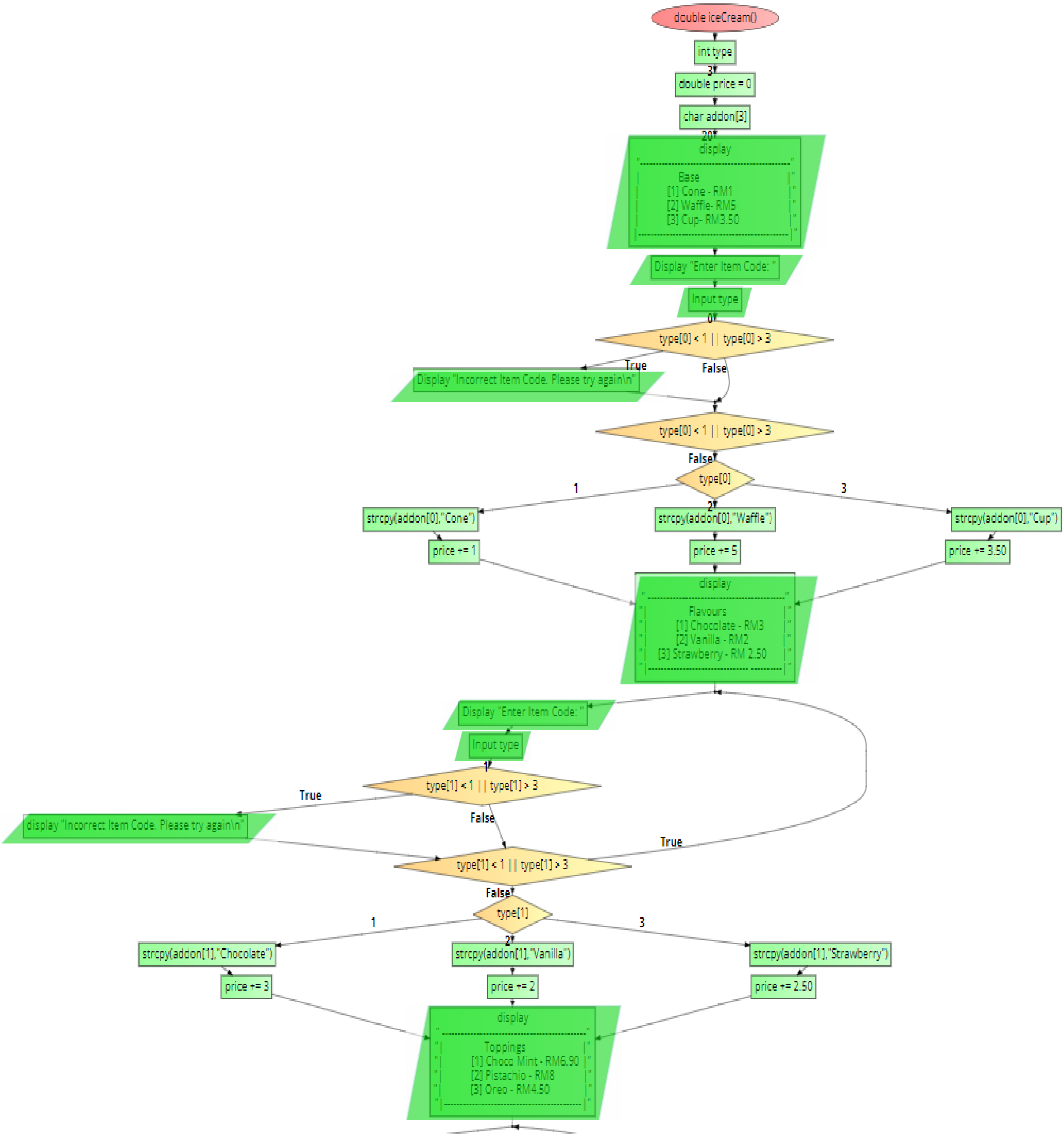


type[1]
1
2
3
strcpy(addon[1],"Chicken") price += 3.50
strcpy(addon[1],"Beef") price += 6
strcpy(addon[1],"Turkey") price += 8
Cheese
[1] Cheddar - RM2.99
[2] Mozzarella - RM3.80
[3] Gorgonzola- RM4
"Enter Item Code: "
type
2
type[2] < 1 || type[2] > 3
True
False
"Incorrect Item Code. Please try again\n"
True
type[2] < 1 || type[2] > 3
False
type[2]
1
2
3
strcpy(addon[2],"Cheeddar") price += 2.99
strcpy(addon[2],"Mozzarella") price += 3.80
strcpy(addon[2],"Gorgonzola") price += 4
SAUCE
[1] Thousand Island - RM3
[2] BBQ- RM1.99
[3] Tomato sauce - RM1
[4] Mayonnaise - RM2.50
[5] Mustard - RM3
"Enter Item Code: "
type
3
type[3] < 1 || type[3] > 5
True
False
"Incorrect Item Code. Please try again\n"
True
type[3] < 1 || type[3] > 5
False

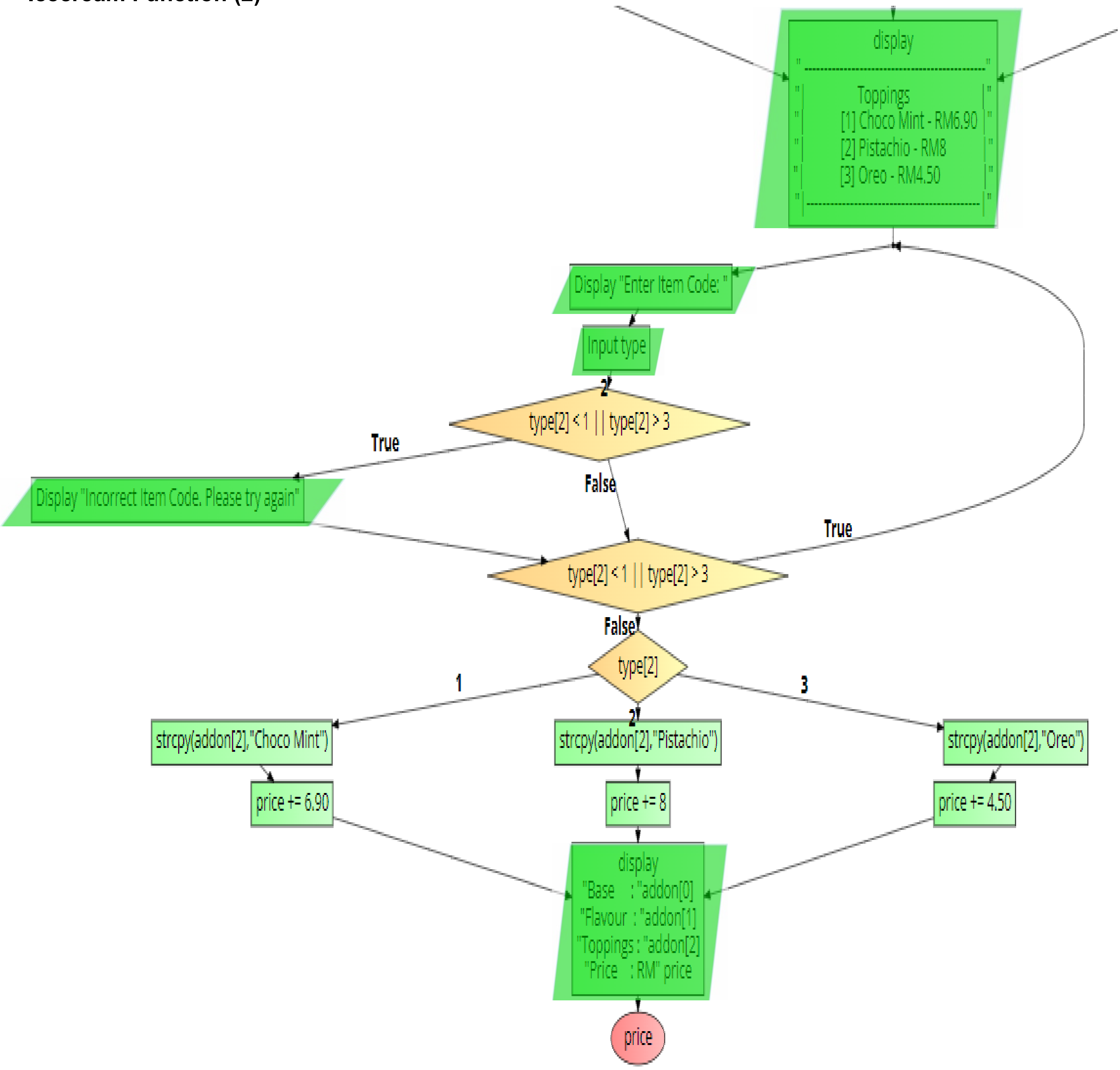
Sandwich Function(3)


type[3] < 1 || type[3] > 5
False
type[3]
1
2
3
4
5
strcpy(addon[3],"Thousand Island") price += 3
strcpy(addon[3],"BBQ") price += 1.99
strcpy(addon[3],"Tomato") price += 1
strcpy(addon[3],"Mayonnaise") price += 2.50
strcpy(addon[3],"Mustard") price +=
"\nBread : "addon[0]
"\nPatty : "addon[1]
"\ncheese : "addon[2]
"\nsauce : "addon[3]
"\nPrice : RM"price
return price

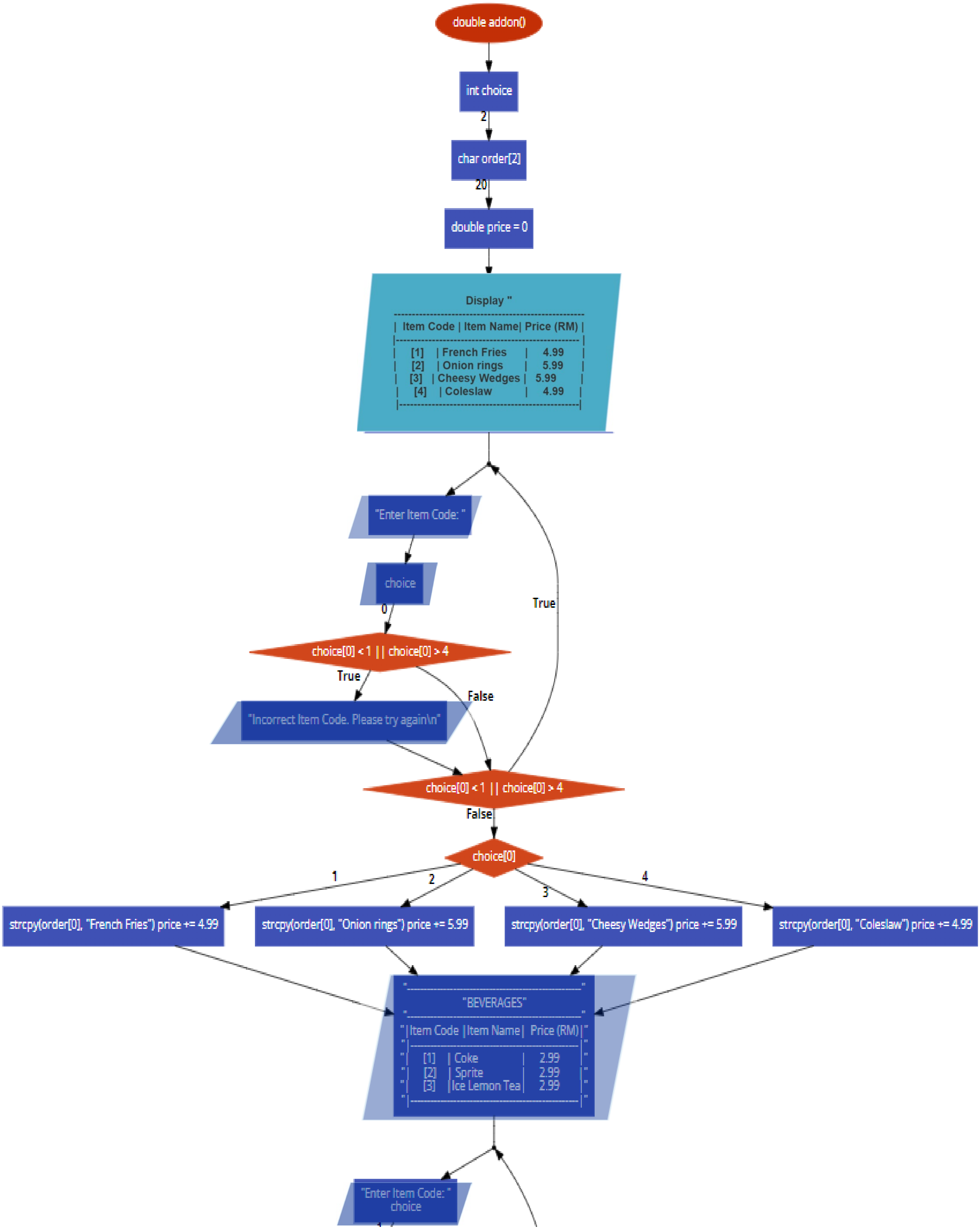
6. Icecream Function (1)



Icecream Function (2)



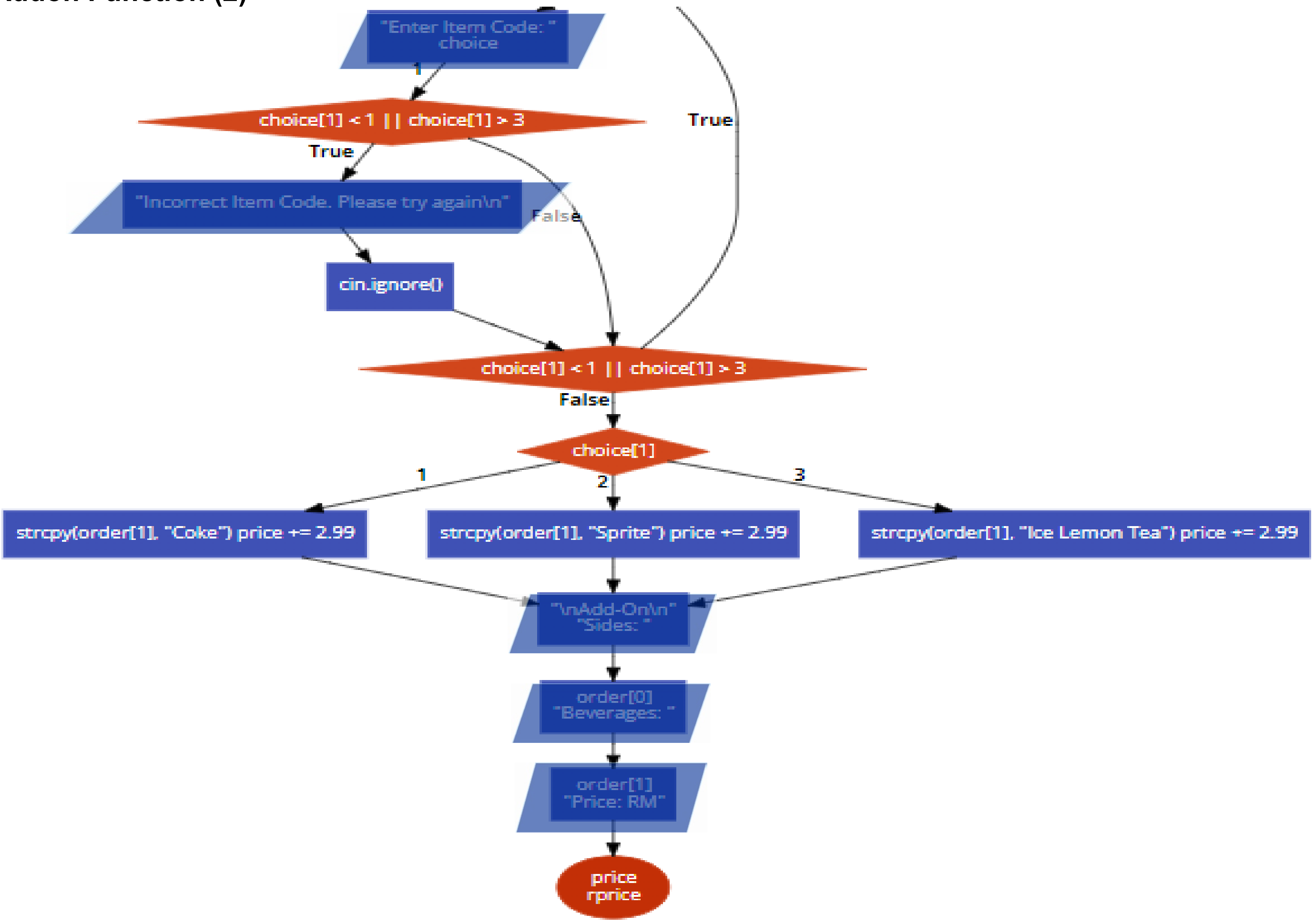
7. Addon Function (1)



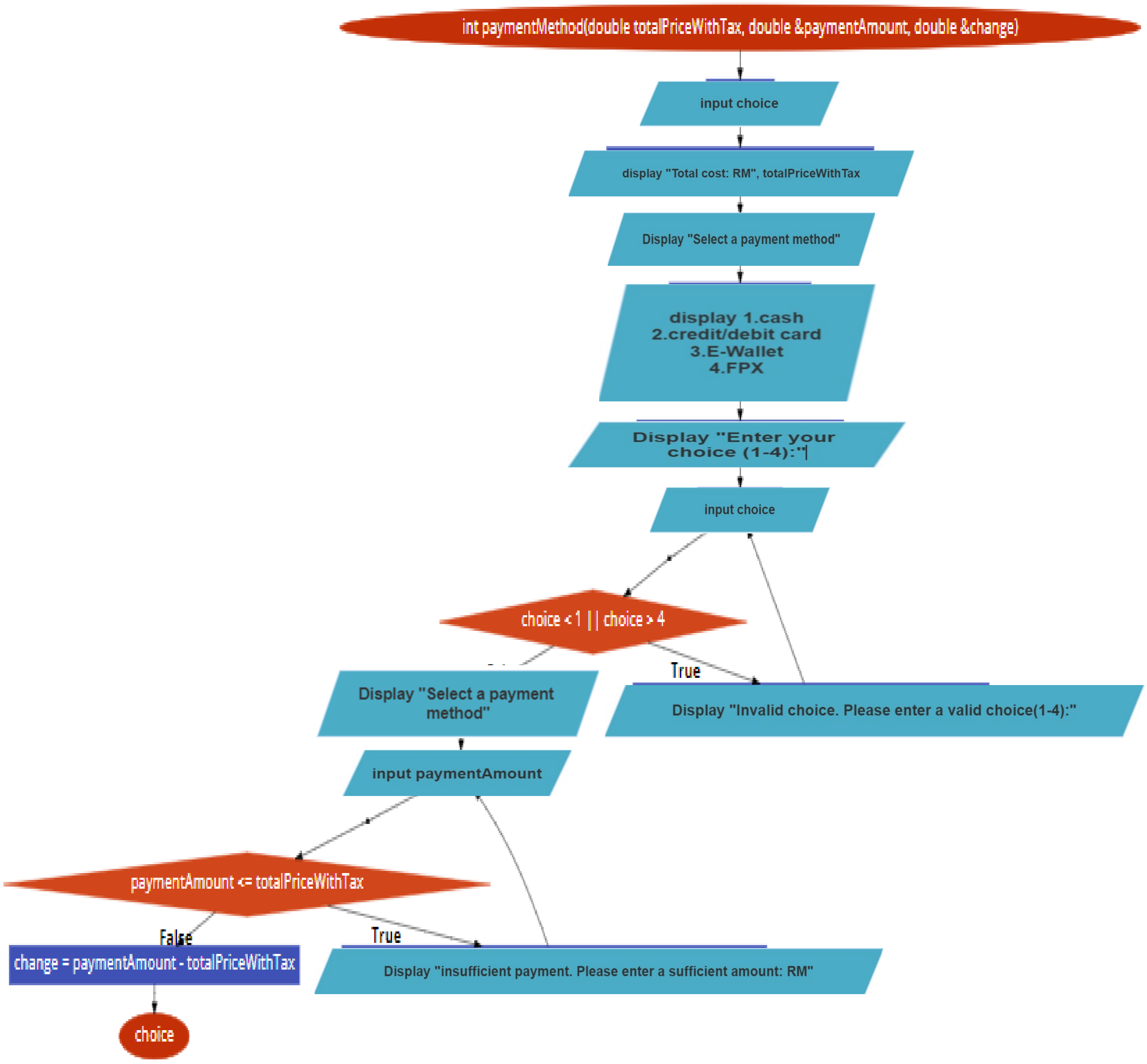
double addon()
int choice
2
char order[2]
20
double price = 0
Display "

Item Code	Item Name	Price (RM)
[1]	French Fries	4.99
[2]	Onion rings	5.99
[3]	Cheesy Wedges	5.99
[4]	Coleslaw	4.99
"Enter Item Code: "
choice
0
choice[0] < 1 || choice[0] > 4
True
"Incorrect Item Code. Please try again\n"
False
True
choice[0] < 1 || choice[0] > 4
False
choice[0]
1
2
3
4
strcpy(order[0], "French Fries") price += 4.99
strcpy(order[0], "Onion rings") price += 5.99
strcpy(order[0], "Cheesy Wedges") price += 5.99
strcpy(order[0], "Coleslaw") price += 4.99
"---"
"BEVERAGES"
"---"
"|Item Code |Item Name| Price (RM)|"
"|---|"
"| [1] | Coke | 2.99 |"
"| [2] | Sprite | 2.99 |"
"| [3] |Ice Lemon Tea| 2.99 |"
"|---|"
"Enter Item Code: "
choice

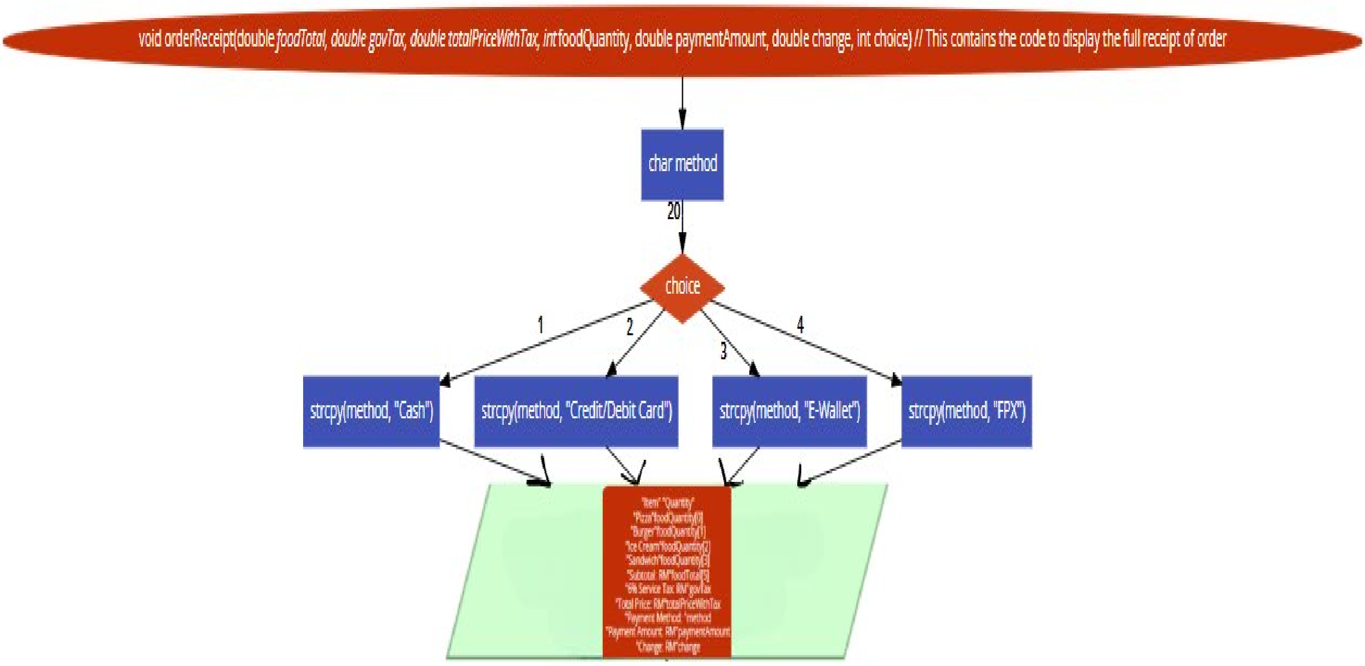
Addon Function (2)


"Enter Item Code: "
choice
1
choice[1] < 1 || choice[1] > 3
True
True
"Incorrect Item Code. Please try again\n"
False
cin.ignore()
choice[1] < 1 || choice[1] > 3
False
choice[1]
1
2
3
strcpy(order[1], "Coke") price += 2.99
strcpy(order[1], "Sprite") price += 2.99
strcpy(order[1], "Ice Lemon Tea") price += 2.99
"\nAdd-On\n"
"Sides: "
order[0]
"Beverages: "
order[1]
"Price: RM"
price
rprice

8. PaymentMethod Function


int paymentMethod(double totalPriceWithTax, double &paymentAmount, double &change)
input choice
display "Total cost: RM", totalPriceWithTax
Display "Select a payment method"
display 1.cash
2.credit/debit card
3.E-Wallet
4.FPX
Display "Enter your choice (1-4):"
input choice
choice < 1 || choice > 4
True
Display "Select a payment method"
Display "Invalid choice. Please enter a valid choice(1-4):"
input paymentAmount
paymentAmount <= totalPriceWithTax
False
True
change = paymentAmount - totalPriceWithTax
Display "insufficient payment. Please enter a sufficient amount: RM"
choice

9. orderReceipt Function


void orderReceipt(double foodTotal, double govTax, double totalPriceWithTax, int foodQuantity, double paymentAmount, double change, int choice) // This contains the code to display the full receipt of order
char method
20
choice
1
2
3
4
strcpy(method, "Cash")
strcpy(method, "Credit/Debit Card")
strcpy(method, "E-Wallet")
strcpy(method, "FPX")
"Item" "Quantity"
"Pizza"foodQuantity[0]
"Burger"foodQuantity[1]
"Ice Cream"foodQuantity[2]
"Sandwich"foodQuantity[3]
"Subtotal: RM"foodTotal[5]
"6% Service Tax: RM"govTax
"Total Price: RM"totalPriceWithTax
"Payment Method: "method
"Payment Amount: RM"paymentAmount
"Change: RM"change

Pseudocode
1.Main Function

```
FUNCTION main()
   DECLARE foodQuantity[4] AS ARRAY OF INTEGER
   DECLARE people, choice AS INTEGER
   DECLARE foodTotal[5] AS ARRAY OF DOUBLE
   DECLARE paymentAmount, change, govTax, totalPriceWithTax AS DOUBLE
   DECLARE food[5] AS DOUBLE

   DISPLAY "How many people would like to order: "
   READ people

   FOR i equal to 0 i less then people  i equals i + 1
      itemCode equal to displayMenu()

      SWITCH itemCode
         CASE 1:
            food[0] equal to pizza()
            foodTotal[0] equal to foodTotal[0] + food[0]
            foodQuantity[0] equal to foodQuantity[0] + 1
            BREAK
         CASE 2:
            food[1] equal to burger()
            foodTotal[1] equal to foodTotal[1] + food[1]
            foodQuantity[1] equal to foodQuantity[1] + 1
            BREAK
         CASE 3:
            food[2] equal to iceCream()
            foodTotal[2] equal to foodTotal[2] + food[2]
            foodQuantity[2] equal to foodQuantity[2] + 1
            BREAK
         CASE 4:
```

```
        food[3] equal to sandwich()
        foodTotal[3] equal to foodTotal[3] + food[3]
        foodQuantity[3] equal to foodQuantity[3] + 1
        BREAK
    END SWITCH

    DISPLAY "Would you like addons that go with your food? (1 for Yes, 2 for No): "
    READ itemCode

    SWITCH itemCode
      CASE 1:
        food[4] equal to addon()
        foodTotal[4] equal to foodTotal[4] + food[4]
        BREAK
    END SWITCH
  END FOR

  foodTotal[5] equal to foodTotal[0] + foodTotal[1] + foodTotal[2] + foodTotal[3] + foodTotal[4]
  govTax equal to foodTotal[5] * 6/100
  totalPriceWithTax equal to foodTotal[5] + govTax

  choice equal to paymentMethod(totalPriceWithTax, paymentAmount, change)
  orderReceipt(foodTotal, govTax, totalPriceWithTax, foodQuantity, paymentAmount, change, choice)
END FUNCTION
```

2. DisplayMenu Function

```
FUNCTION displayMenu()
  DECLARE itemCode AS INTEGER

  DISPLAY "-------------------------------------------------------"
  DISPLAY "          RESTAURANT MENU"
  DISPLAY "-------------------------------------------------------"
  DISPLAY "-------------------------------------------------------"
  DISPLAY "| Item Code |       Item Name       | Base Price (RM)|"
  DISPLAY "-------------------------------------------------------"
  DISPLAY "|    [1]    | Pizza                 |     21.87        |"
  DISPLAY "|    [2]    | Burger                |     2.99         |"
  DISPLAY "|    [3]    | Ice Cream             |      4.50        |"
  DISPLAY "|    [4]    | Sandwich              |     9.49         |"
  DISPLAY "-----------------------------------------------------"

  DO
    DISPLAY "Please enter item code to order: "
    READ itemCode

    IF itemCode less then 0 OR itemCode more then 4 THEN
      DISPLAY "Incorrect item code. Please try again."
      IGNORE next line of input
    END IF

  WHILE itemCode less then 0 OR itemCode more then 4

  RETURN itemCode
END FUNCTION
```

3. Pizza Function

```
FUNCTION pizza()
   DECLARE choice[4] AS ARRAY OF INTEGER
   DECLARE addon[4][20] AS ARRAY OF STRING
   DECLARE price AS DOUBLE

   DISPLAY "--------------------------------------------------"
   DISPLAY "          PIZZA SIZES"
   DISPLAY "--------------------------------------------------"
   DISPLAY "|  Item Code |      Item Name      |   Price (RM)   |"
   DISPLAY "|-----------------------------------------------------------|"
   DISPLAY "|     [1]    | Small              |     13.99      |"
   DISPLAY "|     [2]    | Medium             |     16.99      |"
   DISPLAY "|     [3]    | Large              |     19.99      |"
   DISPLAY "|     [4]    | Extra-Large        |     22.99      |"
   DISPLAY "|--------------------------------------------------|"

   DO
      DISPLAY "Enter Item Code: "
      READ choice[0]

      IF choice[0] less then 1 OR choice[0] more then 4 THEN
         DISPLAY "Incorrect Item Code. Please try again"
         IGNORE next line of input
      END IF

   WHILE choice[0] less then 1 OR choice[0] more then 4

   SWITCH choice[0]
      CASE 1:
         COPY "Small" TO addon[0]
         price equal to price + 13.99
         BREAK
      CASE 2:
         COPY "Medium" TO addon[0]
         price equal to price + 16.99
         BREAK
      CASE 3:
         COPY "Large" TO addon[0]
         price equal to price + 19.99
         BREAK
      CASE 4:
         COPY "Extra-Large" TO addon[0]
         price equal to price + 22.99
         BREAK
   END SWITCH

   DISPLAY "--------------------------------------------------"
   DISPLAY "          PIZZA CRUST"
   DISPLAY "--------------------------------------------------"
   DISPLAY "|  Item Code |      Item Name      |   Price (RM)   |"
   DISPLAY "|--------------------------------------------------|"
   DISPLAY "|     [1]    | Thin Crust          |      1.99       |"
   DISPLAY "|     [2]    | Thick Crust         |     3.99       |"
   DISPLAY "|     [3]    | Flat Crust          |     2.99       |"
   DISPLAY "|--------------------------------------------------|"

   DO
      DISPLAY "Enter Item Code: "
      READ choice[1]

      IF choice[1] less then 1 OR choice[1] more then 3 THEN
         DISPLAY "Incorrect Item Code. Please try again"
         IGNORE next line of input
      END IF

   WHILE choice[1] less then 1 OR choice[1] more then 3

   SWITCH choice[1]
      CASE 1:
         COPY "Thin Crust" TO addon[1]
         price equal to price + 1.99
         BREAK
      CASE 2:
         COPY "Thick Crust" TO addon[1]
         price equal to price + 3.99
         BREAK
      CASE 3:
         COPY "Flat Crust" TO addon[1]
         price equal to price + 2.99
         BREAK
   END SWITCH

   DISPLAY "--------------------------------------------------"
   DISPLAY "           PIZZA TOPPINGS"
   DISPLAY "--------------------------------------------------"
   DISPLAY "|  Item Code |      Item Name      |  Price (RM)   |"
   DISPLAY "|--------------------------------------------------|"
   DISPLAY "|     [1]    | Pepperoni           |     4.90       |"
   DISPLAY "|     [2]    | Extra Cheese        |     4.90       |"
   DISPLAY "|     [3]    | Mushrooms           |     4.90       |"
   DISPLAY "|     [4]    | Sausage             |     4.90       |"
   DISPLAY "|--------------------------------------------------|"

   DO
      DISPLAY "Enter Item Code: "
      READ choice[2]

      IF choice[2] less then 1 OR choice[2] more then 4 THEN
         DISPLAY "Incorrect Item Code. Please try again"
         IGNORE next line of input
      END IF

   WHILE choice[2] less then 1 OR choice[2] more then 4

   SWITCH choice[2]
      CASE 1:
         COPY "Pepperoni" TO addon[2]
```

```
        price equal to price + 4.90
        BREAK
    CASE 2:
        COPY "Extra Cheese" TO addon[2]
        price equal to price + 4.90
        BREAK
    CASE 3:
        COPY "Mushrooms" TO addon[2]
        price equal to price + 4.90
        BREAK
    CASE 4:
        COPY "Sausage" TO addon[2]
        price equal to price + 4.90
        BREAK
END SWITCH

DISPLAY "--------------------------------------------------"
DISPLAY "            PIZZA SAUCES"
DISPLAY "--------------------------------------------------"
DISPLAY "|  Item Code |      Item Name      |   Price (RM)   |"
DISPLAY "|--------------------------------------------------|"
DISPLAY "|     [1]    | Marinara Sauce      |     0.99       |"
DISPLAY "|     [2]    | BBQ Sauce           |     0.99       |"
DISPLAY "|     [3]    | Buffalo Sauce       |     1.50       |"
DISPLAY "|     [4]    | Alfredo Sauce       |     1.99       |"
DISPLAY "|--------------------------------------------------|"

DO
    DISPLAY "Enter Item Code: "
    READ choice[3]

    IF choice[3] less then 1 OR choice[3] more then 4 THEN
        DISPLAY "Incorrect Item Code. Please try again"
        IGNORE next line of input
    END IF

WHILE choice[3] less then 1 OR choice[3] more then 4

SWITCH choice[3]
    CASE 1:
        COPY "Marinara Sauce" TO addon[3]
        price equal to price + 0.99
        BREAK
    CASE 2:
        COPY "BBQ Sauce" TO addon[3]
        price equal to price + 0.99
        BREAK
    CASE 3:
        COPY "Buffalo Sauce" TO addon[3]
        price equal to price + 1.50
        BREAK
    CASE 4:
        COPY "Alfredo Sauce" TO addon[3]
        price equal to price + 1.99
        BREAK
```

```
   END SWITCH

   DISPLAY "Your pizza"
   DISPLAY "Size: " + addon[0]
   DISPLAY "Crust: " + addon[1]
   DISPLAY "Topping: " + addon[2]
   DISPLAY "Sauce: " + addon[3]
   DISPLAY "Price: RM" + price

   RETURN price
END FUNCTION
```

4. Burger Function

```
FUNCTION burger()
   DECLARE num AS INTEGER
   DECLARE order[6] AS ARRAY OF INTEGER
   DECLARE price AS DOUBLE

   DISPLAY "--------------------------------------------------"
   DISPLAY "          BURGER TOPPINGS"
   DISPLAY "--------------------------------------------------"
   DISPLAY "|  Item Code |      Item Name      |   Price (RM)   |"
   DISPLAY "|------------------------------------------------|"
   DISPLAY "|     [1]    | Bison              |     13.99      |"
   DISPLAY "|     [2]    | Beef               |     7.99       |"
   DISPLAY "|     [3]    | Turkey             |     6.99       |"
   DISPLAY "|     [4]    | Chicken            |     5.99       |"
   DISPLAY "|     [5]    | Vegetable          |     2.99       |"
   DISPLAY "|     [6]    | Egg                |     2.99       |"
   DISPLAY "|------------------------------------------------|"

   DISPLAY "How many toppings would you like: "
   READ num

   DECLARE choice[num] AS ARRAY OF INTEGER

   FOR i equal to 0 i less then num i equals i plus 1
      DO
         DISPLAY "Enter Item Code: "
         READ choice[i]

         IF choice[i] less then 1 OR choice[i] more then 6 THEN
            DISPLAY "Incorrect Item Code. Please try again"
            IGNORE next line of input
         END IF

      WHILE choice[i] less then 1 OR choice[i] more then 6

      SWITCH choice[i]
         CASE 1:
            order[0] equal to order[0] + 1
            price equal to price + 13.99
            BREAK
```

```
        CASE 2:
           order[1] equal to order[1] + 1
           price equal to price + 7.99
           BREAK
        CASE 3:
           order[2] equal to order[2] + 1
           price equal to price + 6.99
           BREAK
        CASE 4:
           order[3] equal to order[3] + 1
           price equal to price + 5.99
           BREAK
        CASE 5:
           order[4] equal to order[4] + 1
           price equal to price + 2.99
           BREAK
        CASE 6:
           order[5] equal to order[5] + 1
           price equal to price + 2.99
           BREAK
     END SWITCH
  END FOR

  DISPLAY "Toppings" + "Quantity"
  DISPLAY "Bison" + order[0]
  DISPLAY "Beef" + order[1]
  DISPLAY "Turkey" + order[2]
  DISPLAY "Chicken" + order[3]
  DISPLAY "Vegetable" + order[4]
  DISPLAY "Egg" + order[5]

  DISPLAY "Burger Price: RM" + price

  RETURN price
END FUNCTION
```

5. Sandwich Function

```
FUNCTION sandwich()
  DECLARE type[4] AS ARRAY OF INTEGER
  DECLARE price AS DOUBLE
  DECLARE addon[4][20] AS ARRAY OF STRING

  DISPLAY " _____________________________________________ "
  DISPLAY "|              BREAD                            |"
  DISPLAY "|          [1] White Bread - RM2                |"
  DISPLAY "|          [2] Garlic Bread- RM3.80             |"
  DISPLAY "|          [3] Baguette- RM7                    |"
  DISPLAY "|          [4] Whole Wheat bread - RM2.50       |"
  DISPLAY "|_____________________________________________|"

  DO
     DISPLAY "Enter Item Code: "
     READ type[0]
```

```
   IF type[0] less then 1 OR type[0] more then 4
      DISPLAY "Incorrect Item Code. Please try again"
      IGNORE next line of input
   END IF

WHILE type[0] less then 1 OR type[0] more then 4

SWITCH type[0]
   CASE 1:
      COPY "White Bread" TO addon[0]
      price equal to price + 2
      BREAK
   CASE 2:
      COPY "Garlic Bread" TO addon[0]
      price equal to price + 3.80
      BREAK
   CASE 3:
      COPY "Baguette" TO addon[0]
      price equal to price + 7
      BREAK
   CASE 4:
      COPY "Whole Wheat Bread" TO addon[0]
      price equal to price + 2.50
      BREAK
END SWITCH

DISPLAY " ____________________________________________"
DISPLAY "|            Meat                  |"
DISPLAY "|        [1] Chicken - RM3.50           |"
DISPLAY "|        [2] Beef- RM6               |"
DISPLAY "|        [3] Turkey - RM8              |"
DISPLAY "|____________________________________________|"

DO
   DISPLAY "Enter Item Code: "
   READ type[1]

   IF type[1] less then 1 OR type[1] more then 3 THEN
      DISPLAY "Incorrect Item Code. Please try again"
      IGNORE next line of input
   END IF

WHILE type[1] less then 1 OR type[1] more then 3

SWITCH type[1]
   CASE 1:
      COPY "Chicken" TO addon[1]
      price equal to price + 3.50
      BREAK
   CASE 2:
      COPY "Beef" TO addon[1]
      price equal to price + 6
      BREAK
   CASE 3:
      COPY "Turkey" TO addon[1]
```

```
        price equal to price + 8
        BREAK
END SWITCH

DISPLAY " _____________________________________________"
DISPLAY "|            Cheese                    |"
DISPLAY "|          [1] Cheddar - RM2.99            |"
DISPLAY "|          [2] Mozzarella - RM3.80          |"
DISPLAY "|          [3] Gorgonzola- RM4              |"
DISPLAY "|_____________________________________________|"

DO
    DISPLAY "Enter Item Code: "
    READ type[2]

    IF type[2] less then 1 OR type[2] more then 3 THEN
        DISPLAY "Incorrect Item Code. Please try again"
        IGNORE next line of input
    END IF

WHILE type[2] less then 1 OR type[2] more then 3

SWITCH type[2]
    CASE 1:
        COPY "Cheddar" TO addon[2]
        price equal to price + 2.99
        BREAK
    CASE 2:
        COPY "Mozzarella" TO addon[2]
        price equal to price + 3.80
        BREAK
    CASE 3:
        COPY "Gorgonzola" TO addon[2]
        price equal to price + 4
        BREAK
END SWITCH


DISPLAY " _____________________________________________"
DISPLAY "|            SAUCE                     |"
DISPLAY "|          [1] Thousand Island - RM3         |"
DISPLAY "|          [2] BBQ- RM1.99                |"
DISPLAY "|          [3] Tomato sauce - RM1            |"
DISPLAY "|          [4] Mayonnaise - RM2.50           |"
DISPLAY "|          [5] Mustard - RM3                |"
DISPLAY "|______________________________________________|"

DO
    DISPLAY "Enter Item Code: "
    READ type[3]

    IF type[3] less then 1 OR type[3] more then 5 THEN
        DISPLAY "Incorrect Item Code. Please try again"
        IGNORE next line of input
    END IF
    WHILE type[3] less then 1 OR type[3] more then 5

    SWITCH type[3]
        CASE 1:
            COPY "Thousand Island" TO addon[3]
            price equal to price + 3
            BREAK
        CASE 2:
            COPY "BBQ" TO addon[3]
            price equal to price + 1.99
            BREAK
        CASE 3:
            COPY "Tomato" TO addon[3]
            price equal to price + 1
            BREAK
        CASE 4:
            COPY "Mayonnaise" TO addon[3]
            price equal to price + 2.50
            BREAK
        CASE 5:
            COPY "Mustard" TO addon[3]
            price equal to price + 3
            BREAK
    END SWITCH

    DISPLAY "Bread  : " + addon[0]
    DISPLAY "Patty  : " + addon[1]
    DISPLAY "Cheese : " + addon[2]
    DISPLAY "Sauce  : " + addon[3]
    DISPLAY "Price  : RM" + price

    RETURN price
END FUNCTION
```

6. Icecream Function

```
function iceCream()
    type[3] equal to {0, 0, 0}
    price equal to 0.0
    addon[3][20] equal to {""}

    Display " _____________________________________________"
    Display "|              Base                            |"
    Display "|          [1] Cone - RM1                      |"
    Display "|          [2] Waffle- RM5                     |"
    Display "|          [3] Cup- RM3.50                     |"
    Display "|______________________________________________|"

    // Get the user's choice for base
    do
        Display "Enter Item Code: "
        Input type[0]

        // Validate the user's choice
    if type[0] less then 1 OR type[0] more then 3
        Display "Incorrect Item Code. Please try again"
        Input type[0]
    end if
while type[0] less then 1 OR type[0] more then 3

// Set the base details based on the user's choice
switch type[0]
    case 1: strcpy(addon[0], "Cone")
            price equal to price + 1
    case 2: strcpy(addon[0], "Waffle")
            price equal to price + 5
    case 3: strcpy(addon[0], "Cup")
            price equal to price + 3.50
end switch

Display " ______________________________________________"
Display "|               Flavours                        |"
Display "|          [1] Chocolate - RM3                  |"
Display "|          [2] Vanilla - RM2                     |"
Display "|          [3] Strawberry - RM2.50              |"
Display "|______________________________________________|"

// Get the user's choice for flavour
do
    Display "Enter Item Code: "
    Input type[1]

    // Validate the user's choice
    if type[1] less then 1 OR type[1] more then 3
        Display "Incorrect Item Code. Please try again"
        Input type[1]
    end if
while type[1] less then 1 OR type[1] more then 3

// Set the flavour details based on the user's choice
switch type[1]
    case 1: strcpy(addon[1], "Chocolate")
            price equal to price + 3
    case 2: strcpy(addon[1], "Vanilla")
            price equal to price + 2
    case 3: strcpy(addon[1], "Strawberry")
            price equal to price + 2.50
end switch

Display " ______________________________________________"
Display "|               Toppings                        |"
Display "|          [1] Choco Mint - RM6.90              |"
Display "|          [2] Pistachio - RM8                 |"
Display "|          [3] Oreo - RM4.50                    |"
Display "|______________________________________________|"

// Get the user's choice for toppings
do
    Display "Enter Item Code: "
```

```
      Input type[2]

      // Validate the user's choice
      if type[2] less then 1 OR type[2] more then 3
         Display "Incorrect Item Code. Please try again"
         Input type[2]
      end if
   while type[2] less then 1 OR type[2] more then 3

   // Set the toppings details based on the user's choice
   switch type[2]
      case 1: strcpy(addon[2], "Choco Mint")
            price equal to price + 6.90
      case 2: strcpy(addon[2], "Pistachio")
            priceequal to- price + 8
      case 3: strcpy(addon[2], "Oreo")
            price equal to price + 4.50
   end switch

   // Display the order details
   Display "Base     : " + addon[0]
   Display "Flavour  : " + addon[1]
   Display "Toppings : " + addon[2]
   Display "Price    : RM" + price
   Return price
end function
```

7. Addon Function

```
function addon()
   // Declare variables and arrays
   Choice[2] equal to {0, 0}
   order[2][20] equal to {""}
   price equal to 0.0

   // Display the side dishes menu
   Display "--------------------------------------------------"
   Display "         SIDES"
   Display "--------------------------------------------------"
   Display "|  Item Code |      Item Name      |  Price (RM)   |"
   Display "|-------------------------------------------------|"
   Display "|     [1]    | French Fries        |     4.99      |"
   Display "|     [2]    | Onion rings         |     5.99      |"
   Display "|     [3]    | Cheesy Wedges       |     5.99      |"
   Display "|     [4]    | Coleslaw            |     4.99      |"
   Display "|-------------------------------------------------|"

   // Get the user's choice for sides
   do
      Display "Enter Item Code: "
      Input choice[0]

      // Validate the user's choice
      if choice[0] less then 1 OR choice[0] more then 4
         Display "Incorrect Item Code. Please try again"
```

```
        Input choice[0]
    end if
while choice[0] less then 1 OR choice[0] more then 4

// Set the order details based on the user's choice
switch choice[0]
    case 1: order[0] equal to "French Fries"
            price equal to price + 4.99
    case 2: order[0] equal to "Onion rings"
            price equal to price + 5.99
    case 3: order[0] equal to "Cheesy Wedges"
            price equal to price + 5.99
    case 4: order[0] equal to "Coleslaw"
            price equal to price + 4.99
end switch

// Display the beverage menu
Display "---------------------------------------------------"
Display "         BEVERAGES"
Display "---------------------------------------------------"
Display "|  Item Code |      Item Name      |   Price (RM)   |"
Display "|-------------------------------------------------|"
Display "|     [1]    | Coke                   |     2.99       |"
Display "|     [2]    | Sprite                 |     2.99       |"
Display "|     [3]    | Ice Lemon Tea       |     2.99       |"
Display "|-------------------------------------------------|"

// Get the user's choice for beverages
do
    Display "Enter Item Code: "
    Input choice[1]

    // Validate the user's choice
    if choice[1] less then 1 OR choice[1] more then 3
        Display "Incorrect Item Code. Please try again"
        Input choice[1]
    end if
while choice[1] less then 1 OR choice[1] more then 3

// Set the order details based on the user's choice
switch choice[1]
    case 1: order[1] equal to "Coke"
            price equal to price + 2.99
    case 2: order[1] equal to "Sprite"
            price equal to price + 2.99
    case 3: order[1] equal to "Ice Lemon Tea"
            price equal to price + 2.99
end switch

// Display the order details
Display "Add-On"
Display "Sides: " + order[0]
Display "Beverages: " + order[1]
Display "Price: RM" + price
Return price
```

```
end function
```

8. PaymentMethod Function

```
function paymentMethod(totalPriceWithTax, paymentAmount, change)
   choice equal to 0

   // Display the total cost with tax
   Display "Total Cost: RM" + totalPriceWithTax

   // Display payment method options and get user's choice
   Display "Select a payment method:"
   Display "1. Cash"
   Display "2. Credit/Debit Card"
   Display "3. E-Wallet"
   Display "4. FPX"
   Display "Enter your choice (1-4):"
   Input choice

   // Validate the user's choice within the valid range
   while choice less then 1 OR choice more then 4
      Display "Invalid choice. Please enter a valid choice (1-4):"
      Input choice

   // Get the payment amount from the user
   Display "Enter the payment amount: RM"
   Input paymentAmount

   // Validate the payment amount to be sufficient
   while paymentAmount less then or equal to totalPriceWithTax
      Display "Insufficient payment. Please enter a sufficient amount: RM"
      Input paymentAmount

   // Calculate the change to be given to the customer
   change equal to  paymentAmount - totalPriceWithTax

   // Return the user's choice of payment method
   Return choice
end function
```

**10.	OrderReceipt Function**

```
function orderReceipt(foodTotal, govTax, totalPriceWithTax, foodQuantity, paymentAmount, change, choice)
   method [20]""

   // Map the payment choice to the corresponding method
   switch choice
      case 1: method equal to "Cash"
      case 2: method equal to "Credit/Debit Card"
      case 3: method equal to "E-Wallet"
      case 4: method equal to "FPX"

   // Display the order receipt
   Display "*****ORDER RECEIPT*****"
   Display "Item" + "Quantity"
   Display "Pizza" + foodQuantity[0]
   Display "Burger" + foodQuantity[1]
```

```
    Display "Ice Cream" + foodQuantity[2]
    Display "Sandwich" + foodQuantity[3]
    Display "Subtotal: RM" + foodTotal[5]
    Display "6% Service Tax: RM" + govTax
    Display "Total Price: RM" + totalPriceWithTax
    Display "Payment Method: " + method
    Display "Payment Amount: RM" + paymentAmount
    Display "Change: RM" + change
end function
```

Source Code:

```
/*
Name                    : SHEIKH ADAM BAJUNID BIN MOHD FAISAL, MUHAMAD AZIM HAFIZI BIN CHE
MAT, MOHAMAD IMAN MUZAKKIR BIN ISMAIL
Student id              : 2023135385, 2023172751, 2023159911
Course                  : CSC126
Group                   : 1A
Due Date                : 21 July 2023
Program title   : Food Ordering System
*/

#include <iostream>
#include <iomanip>
#include <string.h>

int displayMenu(); //Function Prototype to display the menu
double burger(); // Function Prototype to calculate price of burger
double pizza(); //Function Prototype to calculate price of pizza
double iceCream(); //Function Prototype to calculate price of Ice Cream
double sandwich(); //Function Prototype to calculate price of sandwich
double addon(); //Function Prototype to calculate price of addon
int paymentMethod(double, double&, double&); //Function Prototype to ask and receive payment
void orderReceipt(double*, double, double, int*, double, double, int); //Function Prototype to display the receipt
at the end

using namespace std;

int main() //Function Main to call other functions and do calculations
{
  int foodQuantity[4] = {0,0,0,0} , people, choice;
    double foodTotal[5], paymentAmount = 0, change = 0, govTax = 0, totalPriceWithTax = 0, food[5];

  cout<<"How many people would like to order: ";
    cin>>people;
```

```
   for (int i = 0; i < people; i++) //loop for how many people would like to order
  {
       int itemCode = displayMenu(); //Display's the menu

       switch (itemCode)
  {
          case 1: food[0] = pizza() ; foodTotal[0] += food[0]; foodQuantity[0] += 1; break; // Calculation for burger
          case 2: food[1] = burger() ; foodTotal[1] += food[1]; foodQuantity[1] += 1; break;  // Calculation for pizza
          case 3: food[2] = iceCream() ; foodTotal[2] += food[2]; foodQuantity[2] += 1; break; // Calculation for
icecream
          case 4: food[3] = sandwich() ; foodTotal[3] += food[3]; foodQuantity[3] += 1; break; // Calculation for
sandwich
       }

       cout<<"Would you like addon's that go with your food?(1 for Yes 2 for No): ";
       cin>>itemCode;

       switch (itemCode)
       {
            case 1: food[4] = addon() ; foodTotal[4] += food[4]; break; // Calculation for addon
  }

   }


   foodTotal[5] = foodTotal[0] + foodTotal[1] + foodTotal[2] + foodTotal[3] + foodTotal[4]; // Calculation to total
cost all 4 food items and 1 addon
   govTax = foodTotal[5] * 6/100; // Calculate for government tax
   totalPriceWithTax = foodTotal[5] + govTax; // Calculate for price with tax

   choice = paymentMethod(totalPriceWithTax, paymentAmount, change); // Function Calling paymentMethod
   orderReceipt(foodTotal ,govTax, totalPriceWithTax, foodQuantity, paymentAmount, change, choice); //
Function Calling for the Order Receipt


}

int displayMenu() // This contains the entire menu of our food ordering system
{
  int itemCode;

   cout<<"--------------------------------------------------------"<<endl;
   cout<<setw(32)<<"RESTAURANT MENU"<<endl;
   cout<<"--------------------------------------------------------"<<endl;
   cout<<"--------------------------------------------------------"<<endl;
   cout<<"| Item Code |        Item Name       | Base Price (RM)|"<<endl;
   cout<<"--------------------------------------------------------"<<endl;
   cout<<"|    [1]    | Pizza                |     21.87      |"<<endl;
   cout<<"|    [2]    | Burger               |     2.99       |"<<endl;
   cout<<"|    [3]    | Ice Cream              |     4.50       |"<<endl;
   cout<<"|    [4]    | Sandwich               |     9.49       |"<<endl;
   cout<<"--------------------------------------------------------"<<endl;

   do
```

```
    {
            cout<<"Please enter item code to order: ";
            cin>>itemCode;

            if (itemCode < 0 || itemCode > 4)
            {
                    cout<<"Incorrect item code. Please try again."<<endl;
                    cin.ignore();
  }

    }while (itemCode < 0 || itemCode > 4);

  return itemCode;
}
double pizza() // This contains the code to build a pizza
{
  int choice [4];
  char addon[4][20];
  double price = 0;
  cout<<"----------------------------------------------------"<<endl;
  cout<<setw(32)<<"PIZZA SIZES"<<endl;
  cout<<"----------------------------------------------------"<<endl;
  cout<<"|  Item Code |      Item Name      |   Price (RM)   |"<<endl;
  cout<<"|---------------------------------------------------|"<<endl;
  cout<<"|     [1]    | Small              |     13.99       |"<<endl;
  cout<<"|     [2]    | Medium              |     16.99    |"<<endl;
  cout<<"|     [3]    | Large              |     19.99       |"<<endl;
  cout<<"|     [4]    | Extra-Large         |     22.99   |"<<endl;
  cout<<"|---------------------------------------------------|"<<endl;

  do
  {
  cout<<"Enter Item Code: ";
  cin>>choice[0];

  if( choice[0] < 1 || choice[0] > 4)
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(choice[0] < 1 || choice[0] > 4 );

  switch(choice[0])
  {
  case 1 : strcpy(addon[0], "Small");price += 13.99; break;
  case 2 : strcpy(addon[0], "Medium");price += 16.99; break;
  case 3 : strcpy(addon[0], "Large");price += 19.99; break;
  case 4 : strcpy(addon[0], "Extra-Large");price += 22.99; break;
  }
  cout<<"----------------------------------------------------"<<endl;
  cout<<setw(32)<<"PIZZA CRUST"<<endl;
  cout<<"----------------------------------------------------"<<endl;
  cout<<"|  Item Code |      Item Name      |   Price (RM)   |"<<endl;
  cout<<"|---------------------------------------------------|"<<endl;
```

```
cout<<"|     [1]    | Thin Crust          |     1.99       |"<<endl;
cout<<"|     [2]    | Thick Crust         |     3.99      |"<<endl;
cout<<"|     [3]    | Flat Crust          |     2.99       |"<<endl;
cout<<"|------------------------------------------------|"<<endl;

do
{
cout<<"Enter Item Code: ";
cin>>choice[1];

if( choice[1] < 1 || choice[1] > 3)
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}

}while(choice[1] < 1 || choice[1] > 3 );

switch(choice[1])
{
case 1 : strcpy(addon[1], "Thin Crust"); price += 1.99; break;
case 2 : strcpy(addon[1],"Thick Crust"); price += 3.99; break;
case 3 : strcpy(addon[1], "Flat Crust"); price += 2.99; break;
}
cout<<"--------------------------------------------------"<<endl;
cout<<setw(32)<<"PIZZA TOPPINGS"<<endl;
cout<<"--------------------------------------------------"<<endl;
cout<<"|  Item Code |      Item Name      |   Price (RM)   |"<<endl;
cout<<"|------------------------------------------------|"<<endl;
cout<<"|     [1]    | Pepperoni           |     4.90       |"<<endl;
cout<<"|     [2]    | Extra Cheese        |     4.90    |"<<endl;
cout<<"|     [3]    | Mushrooms           |     4.90    |"<<endl;
cout<<"|     [4]    | Sausage             |     4.90       |"<<endl;
cout<<"|------------------------------------------------|"<<endl;

do
{
cout<<"Enter Item Code: ";
cin>>choice[2];

if( choice[2] < 1 || choice[2] > 4)
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}

}while(choice[2] < 1 || choice[2] > 4 );

switch(choice[2])
{
case 1 : strcpy(addon[2], "Pepperoni"); price += 4.90; break;
case 2 : strcpy(addon[2], "Extra Cheese"); price += 4.90; break;
case 3 : strcpy(addon[2], "Mushrooms"); price += 4.90; break;
case 4 : strcpy(addon[2], "Sausage"); price += 4.90; break;
}
```

```
  cout<<"--------------------------------------------------"<<endl;
  cout<<setw(32)<<"PIZZA SAUCES"<<endl;
  cout<<"--------------------------------------------------"<<endl;
  cout<<"|  Item Code |      Item Name      |   Price (RM)   |"<<endl;
  cout<<"|-------------------------------------------------|"<<endl;
  cout<<"|     [1]    | Marinara Sauce      |     0.99       |"<<endl;
  cout<<"|     [2]    | BBQ Sauce           |     0.99       |"<<endl;
  cout<<"|     [3]    | Buffalo Sauce       |     1.50       |"<<endl;
  cout<<"|     [4]    | Alfredo Sauce       |     1.99       |"<<endl;
  cout<<"|-------------------------------------------------|"<<endl;

  do
  {
  cout<<"Enter Item Code: ";
  cin>>choice[3];

  if( choice[3] < 1 || choice[3] > 4)
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(choice[3] < 1 || choice[3] > 4 );

  switch(choice[3])
  {
  case 1 : strcpy(addon[3], "Marinara Sauce"); price += 0.99; break;
  case 2 : strcpy(addon[3], "BBQ Sauce"); price += 0.99; break;
  case 3 : strcpy(addon[3], "Buffalo Sauce"); price += 1.50; break;
  case 4 : strcpy(addon[3], "Alfredo Sauce"); price += 1.99; break;
  }
  cout<<"\nYour pizza\n";
  cout<<"Size: "<<addon[0]<<endl;
  cout<<"Crust: "<<addon[1]<<endl;
  cout<<"Topping: "<<addon[2]<<endl;
  cout<<"Sauce: "<<addon[3]<<endl;
  cout<<"Price: RM"<<price<<endl;
  return price;
}

double burger() // This contains the code to build a burger
{
  int num, order[6] = {0,0,0,0,0,0};
  double price = 0;
  cout<<"--------------------------------------------------"<<endl;
  cout<<setw(29)<<"BURGER TOPPINGS"<<endl;
  cout<<"--------------------------------------------------"<<endl;
  cout<<"|  Item Code |      Item Name      |   Price (RM)   |"<<endl;
  cout<<"|-------------------------------------------------|"<<endl;
  cout<<"|     [1]    | Bison               |     13.99      |"<<endl;
  cout<<"|     [2]    | Beef                |     7.99       |"<<endl;
  cout<<"|     [3]    | Turkey              |     6.99       |"<<endl;
  cout<<"|     [4]    | Chicken             |     5.99       |"<<endl;
  cout<<"|     [5]    | Vegetable           |     2.99       |"<<endl;
  cout<<"|     [6]    | Egg                 |     2.99       |"<<endl;
  cout<<"|--------------------------------------------------|"<<endl;

  cout<<"How many toppings would you like: ";
  cin>>num;

  int choice[num];
  for (int i = 0; i < num ; ++i)
  {
  do
  {

  cout<<"Enter Item Code: \n";
  cin>>choice[i];

  if( choice[i] < 1 || choice[i] > 6)
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(choice[i] < 1 || choice[i] > 6 );

  switch(choice[i])
  {
  case 1 : order[0] += 1; price += 13.99; break;
  case 2 : order[1] += 1; price += 7.99; break;
  case 3 : order[2] += 1; price += 6.99; break;
  case 4 : order[3] += 1; price += 5.99; break;
  case 5 : order[4] += 1; price += 2.99; break;
  case 6 : order[5] += 1; price += 2.99; break;
  }
  }
  cout<<setw(10)<<"Toppings"<<setw(10)<<"Quantity"<<endl;
  cout<<setw(10)<<"Bison"<<setw(5)<<order[0]<<endl;
  cout<<setw(10)<<"Beef"<<setw(5)<<order[1]<<endl;
  cout<<setw(10)<<"Turkey"<<setw(5)<<order[2]<<endl;
  cout<<setw(10)<<"Chicken"<<setw(5)<<order[3]<<endl;
  cout<<setw(10)<<"Vegetable"<<setw(5)<<order[4]<<endl;
  cout<<setw(10)<<"Egg"<<setw(5)<<order[5]<<endl;
  cout<<"Burger Price: RM"<<price<<fixed<<setprecision(2)<<endl;
  return price;
}

double iceCream() // This contains the code to build a ice cream
{
  int type[3];
  double price = 0;
  char addon[3][20];

  cout<<" __________________________________________________"<<endl;
  cout<<"|            Base                         |"<<endl;
  cout<<"|          [1] Cone - RM1                   |"<<endl;
  cout<<"|          [2] Waffle- RM5                  |"<<endl;
  cout<<"|          [3] Cup- RM3.50                  |"<<endl;
  cout<<"|__________________________________________________|"<<endl;
```

```
do
{
cout<<"Enter Item Code: ";
cin>>type[0];

if( type[0] < 1 || type[0] > 3 )
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}

}while(type[0] < 1 || type[0] > 3 );

switch(type[0])
{
case 1 : strcpy(addon[0],"Cone"); price += 1 ; break;
case 2 : strcpy(addon[0],"Waffle"); price += 5 ; break;
case 3 : strcpy(addon[0],"Cup"); price += 3.50 ; break;
}

cout<<" ________________________________________________"<<endl;
cout<<"|              Flavours                    |"<<endl;
cout<<"|          [1] Chocolate - RM3              |"<<endl;
cout<<"|          [2] Vanilla - RM2                |"<<endl;
cout<<"|          [3] Strawberry - RM 2.50         |"<<endl;
cout<<"|________________________________________________|"<<endl;

do
{
cout<<"Enter Item Code: ";
cin>>type[1];

if( type[1] < 1 || type[1] > 3 )
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}

}while(type[1] < 1 || type[1] > 3 );

switch(type[1])
{
case 1 : strcpy(addon[1],"Chocolate"); price += 3; break;
case 2 : strcpy(addon[1],"Vanilla"); price += 2; break;
case 3 : strcpy(addon[1],"Strawberry"); price += 2.50; break;
}

cout<<" ________________________________________________"<<endl;
cout<<"|              Toppings                    |"<<endl;
cout<<"|          [1] Choco Mint - RM6.90          |"<<endl;
cout<<"|          [2] Pistachio - RM8              |"<<endl;
cout<<"|          [3] Oreo - RM4.50                |"<<endl;
cout<<"|________________________________________________|"<<endl;
```

```
  do
  {
  cout<<"Enter Item Code: ";
  cin>>type[2];

  if( type[2] < 1 || type[2] > 3 )
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(type[2] < 1 || type[2] > 3 );

  switch(type[2])
  {
  case 1 : strcpy(addon[2],"Choco Mint"); price += 6.90; break;
  case 2 : strcpy(addon[2],"Pistachio"); price += 8; break;
  case 3 : strcpy(addon[2],"Oreo"); price += 4.50; break;
  }

  cout<<"\nBase     : "<<addon[0];
  cout<<"\nFlavour  : "<<addon[1];
  cout<<"\nToppings : "<<addon[2];
  cout<<"\nPrice    : RM"<<price<<endl;

  return price;

}

double sandwich() // This contains the code to build a sandwich
{
  int type[4];
  double price = 0;
  char addon[4][20];

  cout<<" _____________________________________________"<<endl;
  cout<<"|            BREAD                      |"<<endl;
  cout<<"|         [1] White Bread - RM2            |"<<endl;
  cout<<"|         [2] Garlic Bread- RM3.80          |"<<endl;
  cout<<"|         [3] Baguette- RM7                |"<<endl;
  cout<<"|         [4] Whole Wheat bread - RM2.50     |"<<endl;
  cout<<"|_____________________________________________|"<<endl;

  do
  {
  cout<<"Enter Item Code: ";
  cin>>type[0];

  if( type[0] < 1 || type[0] > 4 )
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(type[0] < 1 || type[0] > 4 );
```

```
switch(type[0])
{
case 1 : strcpy(addon[0],"White Bread"); price += 2; break;
case 2 : strcpy(addon[0],"Garlic Bread"); price += 3.80; break;
case 3 : strcpy(addon[0],"Baguette"); price += 7; break;
case 4 : strcpy(addon[0],"Whole Wheat Bread"); price += 2.50; break;
}

cout<<" _______________________________________________"<<endl;
cout<<"|              Meat                        |"<<endl;
cout<<"|          [1] Chicken - RM3.50              |"<<endl;
cout<<"|          [2] Beef- RM6                    |"<<endl;
cout<<"|          [3] Turkey - RM8                  |"<<endl;
cout<<"|_______________________________________________|"<<endl;

do
{
cout<<"Enter Item Code: ";
cin>>type[1];

if( type[1] < 1 || type[1] > 3 )
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}

}while(type[1] < 1 || type[1] > 3 );

switch(type[1])
{
case 1 : strcpy(addon[1],"Chicken"); price += 3.50; break;
case 2 : strcpy(addon[1],"Beef"); price += 6; break;
case 3 : strcpy(addon[1],"Turkey"); price += 8; break;
}


  cout<<" _______________________________________________"<<endl;
cout<<"|             Cheese                        |"<<endl;
cout<<"|          [1] Cheddar - RM2.99             |"<<endl;
cout<<"|          [2] Mozzarella - RM3.80          |"<<endl;
cout<<"|          [3] Gorgonzola- RM4              |"<<endl;
cout<<"|______________________________________________|"<<endl;

do
{
cout<<"Enter Item Code: ";
cin>>type[2];

if( type[2] < 1 || type[2] > 3 )
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}
```

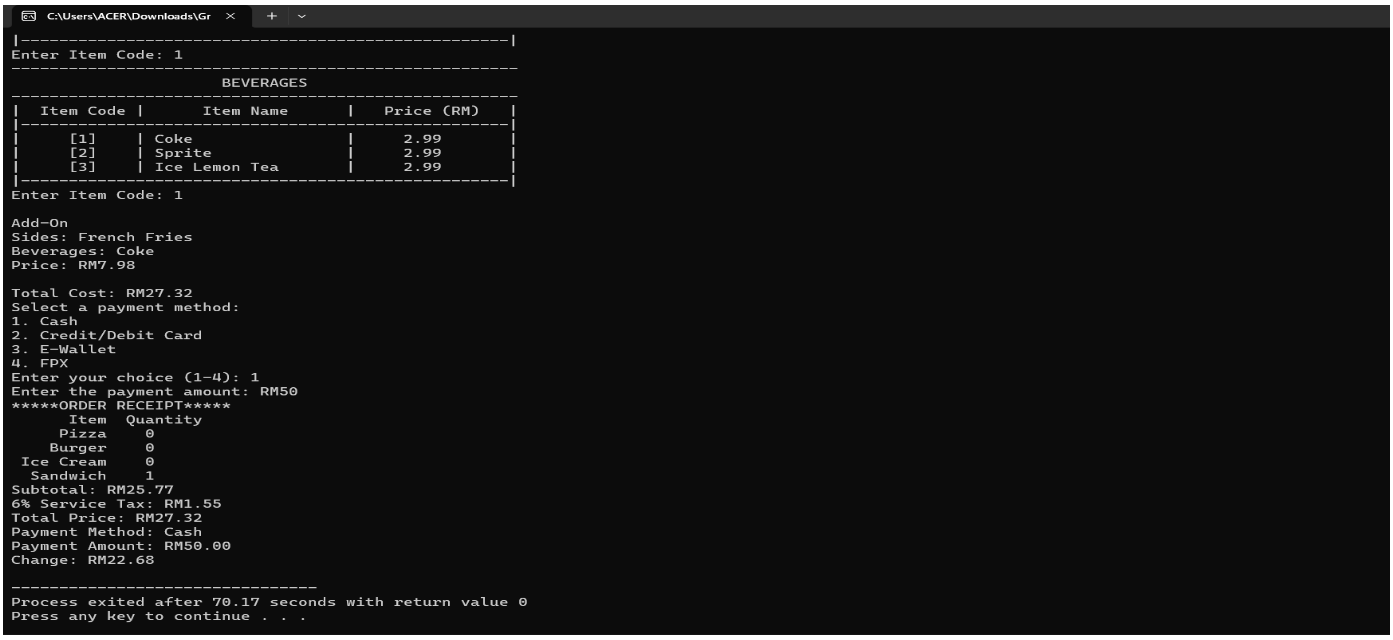
```
  }while(type[2] < 1 || type[2] > 3 );

  switch(type[2])
  {
  case 1 : strcpy(addon[2],"Cheeddar"); price += 2.99; break;
  case 2 : strcpy(addon[2],"Mozzarella"); price += 3.80; break;
  case 3 : strcpy(addon[2],"Gorgonzola"); price += 4; break;
  }


  cout<<" ____________________________________________"<<endl;
  cout<<"|          SAUCE                     |"<<endl;
  cout<<"|        [1] Thousand Island - RM3         |"<<endl;
  cout<<"|        [2] BBQ- RM1.99                 |"<<endl;
  cout<<"|        [3] Tomato sauce - RM1            |"<<endl;
  cout<<"|        [4] Mayonnaise - RM2.50           |"<<endl;
  cout<<"|        [5] Mustard - RM3                |"<<endl;
  cout<<"|____________________________________________|"<<endl;

  do
  {
  cout<<"Enter Item Code: ";
  cin>>type[3];

  if( type[3] < 1 || type[3] > 5 )
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }

  }while(type[3] < 1 || type[3] > 5 );

  switch(type[3])
  {
  case 1 : strcpy(addon[3],"Thousand Island"); price += 3; break;
  case 2 : strcpy(addon[3],"BBQ"); price += 1.99; break;
  case 3 : strcpy(addon[3],"Tomato"); price += 1; break;
  case 4 : strcpy(addon[3],"Mayonnaise"); price += 2.50; break;
  case 5 : strcpy(addon[3],"Mustard"); price += 3; break;
  }



  cout<<"\nBread  : "<<addon[0];
  cout<<"\nPatty  : "<<addon[1];
  cout<<"\ncheese : "<<addon[2];
  cout<<"\nsauce  : "<<addon[3];
  cout<<"\nPrice  : RM"<<price;

  return price;
}

double addon() // This contains the code to build addons for food
{
  int choice[2]; // Declare an array to store the user's choice
```

```
char order[2][20]; // Declare an array to store the order details
double price = 0; // Declare a variable to store the price


// Print the side dishes menu
cout<<"---------------------------------------------------"<<endl;
cout<<setw(29)<<"SIDES"<<endl;
cout<<"---------------------------------------------------"<<endl;
cout<<"|  Item Code |      Item Name      |  Price (RM)   |"<<endl;
cout<<"|-------------------------------------------------|"<<endl;
cout<<"|     [1]    | French Fries        |     4.99       |"<<endl;
cout<<"|     [2]    | Onion rings         |     5.99       |"<<endl;
cout<<"|     [3]    | Cheesy Wedges       |     5.99       |"<<endl;
cout<<"|     [4]    | Coleslaw            |     4.99       |"<<endl;
cout<<"|-------------------------------------------------|"<<endl;


// Get the user's choice for sides
do
{
cout<<"Enter Item Code: ";
cin>>choice[0];


// Validate the user's choice
if( choice[0] < 1 || choice[0] > 4)
{
cout<<"Incorrect Item Code. Please try again\n";
cin.ignore();
}
}while(choice[0] < 1 || choice[0] > 4 );


// Set the order details based on the user's choice
switch(choice[0])
{
case 1 : strcpy(order[0], "French Fries"); price += 4.99; break;
case 2 : strcpy(order[0], "Onion rings"); price += 5.99; break;
case 3 : strcpy(order[0], "Cheesy Wedges"); price += 5.99; break;
case 4 : strcpy(order[0], "Coleslaw"); price += 4.99; break;
}

// Print the beverage menu
cout<<"---------------------------------------------------"<<endl;
cout<<setw(31)<<"BEVERAGES"<<endl;
cout<<"---------------------------------------------------"<<endl;
cout<<"|  Item Code |      Item Name      |  Price (RM)   |"<<endl;
cout<<"|-------------------------------------------------|"<<endl;
cout<<"|     [1]    | Coke                |     2.99       |"<<endl;
cout<<"|     [2]    | Sprite              |     2.99       |"<<endl;
cout<<"|     [3]    | Ice Lemon Tea       |     2.99       |"<<endl;
cout<<"|-------------------------------------------------|"<<endl;


// Get the user's choice for beverages
```

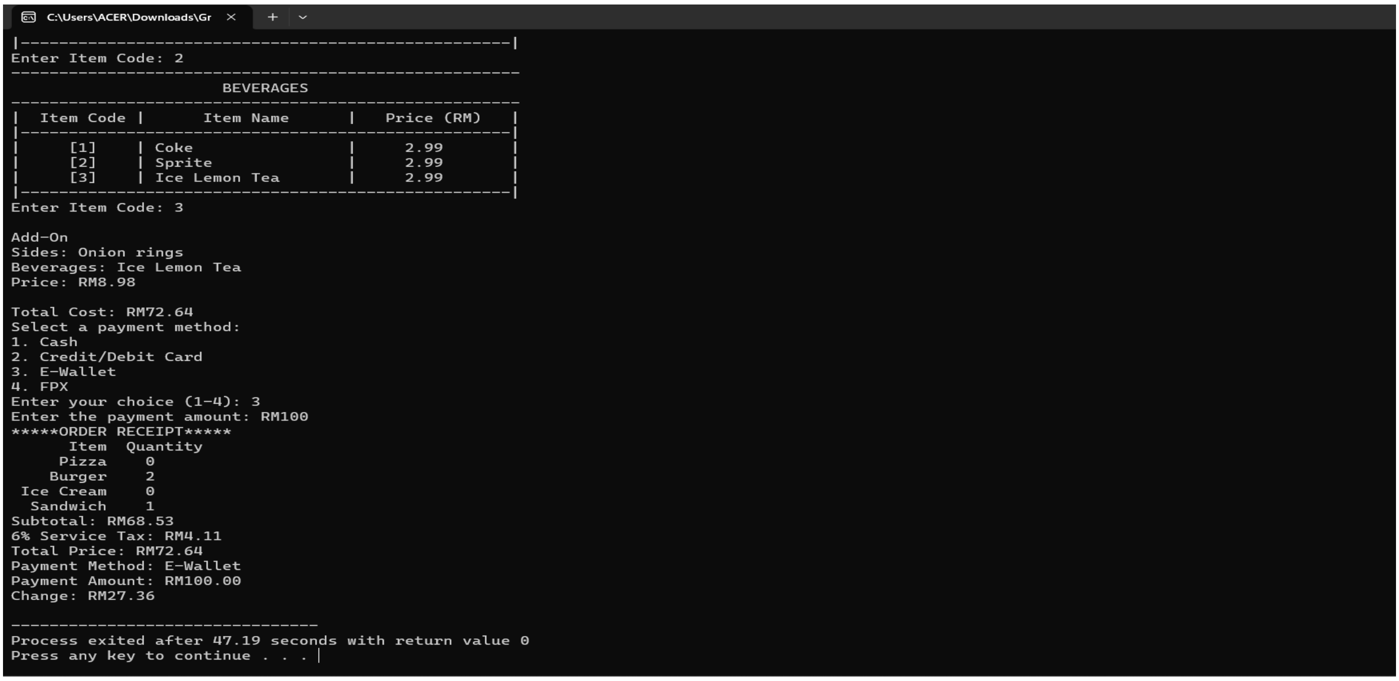
```
  do
  {
  cout<<"Enter Item Code: ";
  cin>>choice[1];

  // Validate the user's choice
  if( choice[1] < 1 || choice[1] > 3)
  {
  cout<<"Incorrect Item Code. Please try again\n";
  cin.ignore();
  }
  }while(choice[1] < 1 || choice[1] > 3 );

  // Set the order details based on the user's choice
  switch(choice[1])
  {
  case 1 : strcpy(order[1], "Coke"); price += 2.99; break;
  case 2 : strcpy(order[1], "Sprite"); price += 2.99; break;
  case 3 : strcpy(order[1], "Ice Lemon Tea"); price += 2.99; break;
  }

  // Print the order details
  cout<<"\nAdd-On\n";
  cout<<"Sides: "<<order[0]<<endl;
  cout<<"Beverages: "<<order[1]<<endl;
  cout<<"Price: RM"<<price<<endl;
  return price;
}

int paymentMethod(double totalPriceWithTax, double &paymentAmount, double &change) // This contains the
code to ask and receive payment
{
  int choice;
  cout<<endl<<"Total Cost: RM"<<fixed<<setprecision(2)<<totalPriceWithTax<<endl;
  cout<<"Select a payment method: "<<endl;
  cout<<"1. Cash"<<endl;
  cout<<"2. Credit/Debit Card"<<endl;
  cout<<"3. E-Wallet"<<endl;
  cout<<"4. FPX"<<endl;
  cout<<"Enter your choice (1-4): ";
  cin>>choice;

  while (choice < 1 || choice > 4)
  {
  cout<<"Invalid choice. Please enter a valid choice (1-4): ";
        cin>>choice;
  }


  cout<<"Enter the payment amount: RM";
  cin>>paymentAmount;

  while (paymentAmount <= totalPriceWithTax)
  {
            cout<<"Insufficient payment. Please enter a sufficient amount: RM";
```

```
          cin>>paymentAmount;
    }

  change = paymentAmount - totalPriceWithTax;

  return choice;
}

void orderReceipt(double* foodTotal, double govTax, double totalPriceWithTax, int* foodQuantity, double
paymentAmount, double change, int choice) // This contains the code to display the full receipt of order
{
  char method[20];

  switch (choice)
  {
  case 1: strcpy(method, "Cash"); break;
  case 2: strcpy(method, "Credit/Debit Card"); break;
  case 3: strcpy(method, "E-Wallet"); break;
  case 4: strcpy(method, "FPX"); break;
  }

  cout<<"*****ORDER RECEIPT*****"<<endl;
  cout<<setw(10)<<"Item"<<setw(10)<<"Quantity"<<endl;
    cout<<setw(10)<<"Pizza"<<setw(5)<<foodQuantity[0]<<endl;
    cout<<setw(10)<<"Burger"<<setw(5)<<foodQuantity[1]<<endl;
    cout<<setw(10)<<"Ice Cream"<<setw(5)<<foodQuantity[2]<<endl;
    cout<<setw(10)<<"Sandwich"<<setw(5)<<foodQuantity[3]<<endl;
    cout<<setw(10)<<"Subtotal: RM"<<foodTotal[5]<<endl;
    cout<<setw(15)<<"6% Service Tax: RM"<<govTax<<fixed<<setprecision(2)<<endl;
    cout<<setw(15)<<"Total Price: RM"<<totalPriceWithTax<<fixed<<setprecision(2)<<endl;
    cout<<setw(15)<<"Payment Method: "<<method<<fixed<<setprecision(2)<<endl;
    cout<<setw(15)<<"Payment Amount: RM"<<paymentAmount<<fixed<<setprecision(2)<<endl;;
    cout<<setw(10)<<"Change: RM"<<change<<fixed<<setprecision(2)<<endl;

}
```

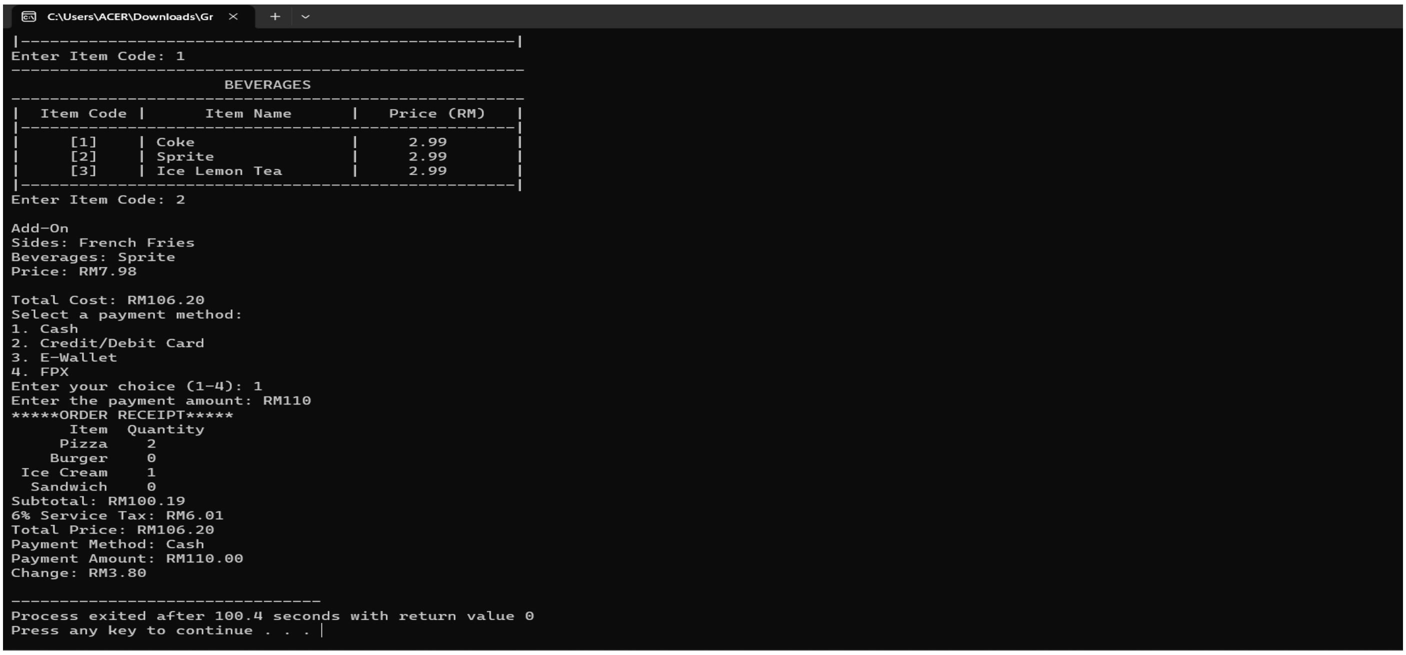
Samples of output:



```
|----------------------------------------------------|
Enter Item Code: 1
-----------------------------------------------------
                      BEVERAGES
-----------------------------------------------------
|  Item Code |       Item Name       |   Price (RM)   |
|----------------------------------------------------|
|     [1]    | Coke                  |     2.99       |
|     [2]    | Sprite                |     2.99       |
|     [3]    | Ice Lemon Tea         |     2.99       |
|----------------------------------------------------|
Enter Item Code: 1

Add-On
Sides: French Fries
Beverages: Coke
Price: RM7.98

Total Cost: RM27.32
Select a payment method:
1. Cash
2. Credit/Debit Card
3. E-Wallet
4. FPX
Enter your choice (1-4): 1
Enter the payment amount: RM50
*****ORDER RECEIPT*****
      Item  Quantity
     Pizza     0
    Burger     0
 Ice Cream     0
  Sandwich     1
Subtotal: RM25.77
6% Service Tax: RM1.55
Total Price: RM27.32
Payment Method: Cash
Payment Amount: RM50.00
Change: RM22.68

--------------------------------
Process exited after 70.17 seconds with return value 0
Press any key to continue . . .
```



```
|----------------------------------------------------|
Enter Item Code: 2
-----------------------------------------------------
                      BEVERAGES
-----------------------------------------------------
|  Item Code |       Item Name       |   Price (RM)   |
|----------------------------------------------------|
|     [1]    | Coke                  |     2.99       |
|     [2]    | Sprite                |     2.99       |
|     [3]    | Ice Lemon Tea         |     2.99       |
|----------------------------------------------------|
Enter Item Code: 3

Add-On
Sides: Onion rings
Beverages: Ice Lemon Tea
Price: RM8.98

Total Cost: RM72.64
Select a payment method:
1. Cash
2. Credit/Debit Card
3. E-Wallet
4. FPX
Enter your choice (1-4): 3
Enter the payment amount: RM100
*****ORDER RECEIPT*****
      Item  Quantity
     Pizza     0
    Burger     2
 Ice Cream     0
  Sandwich     1
Subtotal: RM68.53
6% Service Tax: RM4.11
Total Price: RM72.64
Payment Method: E-Wallet
Payment Amount: RM100.00
Change: RM27.36

--------------------------------
Process exited after 47.19 seconds with return value 0
Press any key to continue . . .
```



```
|---------------------------------------------------|
Enter Item Code: 1
-----------------------------------------------------
                      BEVERAGES
-----------------------------------------------------
|  Item Code |      Item Name      |   Price (RM)   |
|---------------------------------------------------|
|     [1]     | Coke               |     2.99       |
|     [2]     | Sprite             |     2.99       |
|     [3]     | Ice Lemon Tea      |     2.99       |
|---------------------------------------------------|
Enter Item Code: 2

Add-On
Sides: French Fries
Beverages: Sprite
Price: RM7.98

Total Cost: RM106.20
Select a payment method:
1. Cash
2. Credit/Debit Card
3. E-Wallet
4. FPX
Enter your choice (1-4): 1
Enter the payment amount: RM110
*****ORDER RECEIPT*****
      Item  Quantity
     Pizza    2
    Burger    0
 Ice Cream    1
  Sandwich    0
Subtotal: RM100.19
6% Service Tax: RM6.01
Total Price: RM106.20
Payment Method: Cash
Payment Amount: RM110.00
Change: RM3.80

--------------------------------
Process exited after 100.4 seconds with return value 0
Press any key to continue . . .
```

Discussion/Conclusion:

In conclusion, the food ordering system program in C++ has been successfully completed. This program provides a user-friendly interface for customers to place their food orders and for restaurant owners to manage and process those orders efficiently as they no longer have human interaction during the order process which may lead to mistakes such as forgetting an item the customer bought or accidentally choosing the wrong item for them. By implementing key features such as menu selection, order customization, payment processing, and cost, such as government tax calculation, this program streamlines the food ordering process and enhances the overall customer experience.