SUNWAY COLLEGE

COURSEWORK

Program : [X] Diploma in Information Technology
[X] Diploma in Computer Science

Subject : PROGRAMMING CONCEPTS & PROBLEM SOLVING
Subject code : DIT 1253
Due Date : Week 10 (27 October 2023)

NAME	STUDENT ID
SUM JO EE	**23073349**
CHANG KAR CHENG	**23077324**
WONG KAR ERN	**23077142**
TEH SHI YING	**23057235**
VANI KARNAN	**23096506**

SDS ACADEMIC INTEGRITY STATEMENT

Sunway Diploma Studies is committed to the principles of academic integrity. Academic integrity means placing five fundamental values: **honesty, trust, fairness, respect, and responsibility** into practice. It is being honest in the academic work you do at the programme/college, being fair to others, and taking responsibility for learning, and acting in an ethical manner in all your academic endeavours. We believe that these five values are truly foundational to the programme.

We hereby declare that:

1. We fully understand and will uphold the academic integrity of Sunway Diploma Studies (SDS).
2. We confirm that the work hereby submitted is our own original work and where other people's work has been used this has been fully acknowledged.
3. We are aware of the importance of conducting exams with integrity and fairness, and We hereby confirm that We will comply with the requirements.
4. We are aware that non-compliance with these instructions and unfair conduct constitute a disciplinary offense, and actions will be taken including but not limited to being expelled.

No	Name	Signature	Date
1	SUM JO EE		30/10/2023
2	CHANG KAR CHENG		30/10/2023
3	WONG KAR ERN		30/10/2023
4	TEH SHI YING		30/10/2023
5	VANI KARNAN		30/10/2023

Group Member Contribution Form

INSTRUCTION

The contribution must be signed by all members. 3 marks will be deducted from the total marks awarded if the group fail to comply with this requirement

Group Name: __Group 13__

Student Name	**Student ID**	**Signature**	**Role & Responsibilities (e.g. create flow chart, C++ coding etc)**
SUM JO EE	**23073349**		C++ coding and prepare sample output
CHANG KAR CHENG	**23077324**		Create PAC, flowchart and pseudocode
WONG KAR ERN	**23077142**		C++ coding
TEH SHI YING	**23057235**		Create PAC and flow chart
VANI KARNAN	**23096506**		Compile report, create PAC and pseudocode

Marking Scheme

Criteria	Exceeds Requirements	Meets All Requirements	Meets basic Requirements	Marks
	13-15 marks	**10-12 marks**	**0-9 marks**	
Technical Correctness 15%	• No technical, syntax or structure errors • Able to achieve all the anticipated result	• Some technical, syntax or structure errors • Not able to achieve some of the anticipated result	• Many technical, syntax or structure errors • Not able to achieve most or all the anticipated result •	
	6-5 marks	**4-3 marks**	**0-2 marks**	
Problem Solving Logic 6%	• Excellent problem-solving logic • Able to cater for all the scenarios.	• Some deficiency in problem solving logic or problem-solving logic is not optimum	• Many deficiencies in problem solving logic	
	8-6 marks	**5-3 marks**	**0-2 marks**	
Solution Design (PAC & Flow chart) 8%	• Well-structured PAC • The problem well analysed and solution provided accordingly. • Flowchart developed with correct notations and right flow. • Flowchart narrates the program flow completely.	• PAC is structured. • The problem is analysed, and solution is provided. • Flowchart developed with minor errors. • Flowchart doesn't narrate the program flow as how the program is developed.	• PAC is structured • The problem is analysed, and solution is provided. • Flowchart developed with many errors. • Flowchart doesn't narrate the program flow as how the program is developed. •	
	3 marks	**2 marks**	**0-1 mark**	
Comments 3%	• Useful comments to elaborate the meaning of a statement / a block of statements. • Comment place in a proper manner and not overwhelming.	• Contains some comments to elaborate the meaning of a statement / a block of statements. • Comment might not be placed in a proper manner	• Contains little or no comments	

Best practices 3%	- Good programming ethics and practices and implemented many good programming styles.	- Good programming ethics and practices and implemented few good programming styles.	- Poor programming ethics and practices and implemented many good programming styles.	
Comments			**TOTAL (35%)**	

PROBLEM ANALYSIS CHART (PAC)

Given Data	**Required result**
productType kitType vegeProduce package type items monthlyQuantity annualQuantity kitQuantity continueBuying Cash	Printing receipt Salad finger co. Date and time Location Order number Invoice Bill number status total quantity total price(RM) grand total (incl.6%) cash change Thank you , please continue to use our service
Processing required	**Solution Alternatives**
FOR 1 = vegetable subscription FOR 2 = smart kit If input = 1 or 2 productType = vegetable subscription or smart kit Else “Invalid product, pick again!” END IF END FOR Calculate (kitType) If (kitType = 1) then	**None**

```
Output “You’ve picked Farm Kit”
(farm kit)
            totalPrice =kitQuantity*19
            totalItem = kitQuantity
If (kitType = 2) then
Output ”You’ve picked Compost Kit”
(compost kit)
            totalPrice =kitQuantity*49
            totalItem = kitQuantity
 If (kitType = 3) then
Output “You’ve picked Microgreens Kit”
(microgreens kit)
            totalPrice =kitQuantity*35
            totalItem = kitQuantity
END IF
 if (input = "1" or input = "2)
        packageType = monthly grower or annual grower
        else
        Output "Invalid product, pick again!"
END IF
Input = "1" or input = "2" or input = "3" or input = "4" or input = "5"
vegeProduce =  1. Monthly Grower 1 - Leafy or 2. Monthly Grower 2
- Leafy or 3. Monthly Grower 3 - Leafy or 4. Monthly Grower 1 -
Fruity or 5. Monthly Grower 2 - Fruity
            (vegeProduce)
If ( monthly grower 1-fruity ) then
            totalPrice = monthlyQuantity * 120
            totalitem=monthlyQuantity

 If (monthly grower 2-fruity ) then
            totalPrice = monthlyQuantity * 240
            totalitem=monthlyQuantity
```

```
If (monthly grower 3-leafy) then
totalPrice=montlyQuantity * 270
            totalItem=montlyQuantity
END IF
(items)
PRINT “Pick a vegetable produce:
1. Monthly Grower 1 - Leafy, 2. Monthly Grower 2 - Leafy, 3.
Monthly Grower 3 - Leafy, 4. Monthly Grower 1 - Fruity, 5. Monthly
Grower 2 - Fruity”
Switch(vegeProduce)
            Case 1 : “You've picked Monthly Grower 1 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad &
Asian Mix
Enter chosen items:”
totalPrice =monthlyQuantity * 105
                        totalItem =monthlyQuantity,
BREAK
Case 2 : “You've picked Monthly Grower 2 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad &
Asian Mix
Enter chosen items:”
totalPrice=monthlyQuantity *189
                        totalItem = monthlyQuantity,
, BREAK
Case 3 : “You've picked Monthly Grower 3 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad &
Asian Mix
Enter chosen items:”
totalPrice=monthlyQuantity * 270
totalItem=monthlyQuantity,
, BREAK
Case 4 : “You've picked Monthly Grower 1 - Fruity
Pick items:  Red/yellow capsicums, tomatoes/cherry, green/red chilis
english/cocktail cucumbers
Insert quantity:”
totalPrice = monthlyQuantity * 120
totalItem = monthlyQuantity, BREAK
Case 5 : “You've picked Monthly Grower 2 - Fruity
Pick items:  Red/yellow capsicums, tomatoes/cherry, green/red chilis
english/cocktail cucumbers
Insert quantity:”
totalPrice = monthlyQuantity * 240
totalItem = monthlyQuantity, BREAK
END CASE


totalPrice=monthlyQuantity * 270
totalItem=monthlyQuantity
if items >= 1 and items <= 3 or itemException = 1
switch(item)
Case 1 = PRINT “Insert quantity:”
totalPrice = monthlyQuantity * 105
totalItem = monthlyQuantity, BREAK
Case 2 = PRINT “Insert quantity:”
totalPrice = monthlyQuantity * 189
totalItem = monthlyQuantity, BREAK
Case 3 = PRINT “Insert quantity:”
totalPrice += monthlyQuantity * 270
totalItem += monthlyQuantity, BREAK
END CASE
  if (items >= 1 and items <= 3)
PRINT “You've picked Annual Grower 1 - Leafy"
```

```
 "Pick items: 1. salad lettuces, 2. hearty asian greens, 3. salad & asian
mix"
END IF
if (items >= 1 and items <= 3)
switch(items)
Case 1 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
totalItem += annualQuantity, BREAK
Case 2 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
totalItem += annualQuantity, BREAK
Case 3 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
totalItem += annualQuantity, BREAK
END CASE
grandTotal(incl.6%) = (totalPrice * 0.06) + totalPrice
totalChange = cash - grandTotal
If totalItems > 0
Display grandTotal
If cash < grandTotal
Output ( "Amount entered is not sufficient. Please enter amount: RM"
)
Else
Output ( "Thank you for shopping with us! come again!" )
End if
```

PSEUDOCODE

```
BEGIN
        Int productType , packageType , kitType, itemException
        Double totalPrice=0 , grandTotal , cash =0, totalChange
        Int monthlyQuantity , annualQuantity ,items =0,kitQuantity , totalItem=0, oderNum=0, billNum=1234
        Char continueBuying , hypen=45
        odernNum++
        billNum++
         Input
                productType
kitType
vegeProduce
                package type
                items
monthlyQuantity
annualQuantity
                kitQuantity
continueBuying
                cash



      Process
FOR 1 =  vegetable subscription
FOR 2 = smart kit
If input = 1 or 2
productType = vegetable subscription or smart kit
Else "Invalid product, pick again!"
END IF
END FOR
                Calculate
                (kitType)
If (kitType = 1) then
Output "You've picked Farm Kit"
```

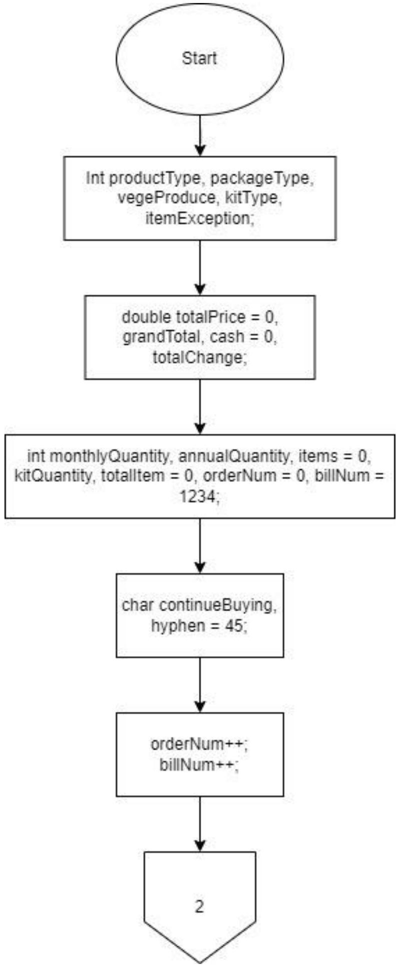
```
(farm kit)
                    totalPrice =kitQuantity*19
                    totalItem = kitQuantity
If (kitType = 2) then
Output ”You’ve picked Compost Kit”
(compost kit)
                    totalPrice =kitQuantity*49
                    totalItem = kitQuantity
 If (kitType = 3) then
Output “You’ve picked Microgreens Kit”
(microgreens kit)
                    totalPrice =kitQuantity*35
                    totalItem = kitQuantity
END IF
 if (input = "1" or input = "2)
             packageType = monthly grower or annual grower
             else
             Output "Invalid product, pick again!"
END IF
Input = "1" or input = "2" or input = "3" or input = "4" or input = "5"
vegeProduce =  1. Monthly Grower 1 - Leafy or 2. Monthly Grower 2 - Leafy or 3. Monthly Grower 3 - Leafy or 4. Monthly Grower 1 - Fruity
or 5. Monthly Grower 2 - Fruity
                    (vegeProduce)
If ( monthly grower 1-fruity ) then
                    totalPrice = monthlyQuantity * 120
                    totalitem=monthlyQuantity

 If (monthly grower 2-fruity ) then
                    totalPrice = monthlyQuantity * 240
                    totalitem=monthlyQuantity
```

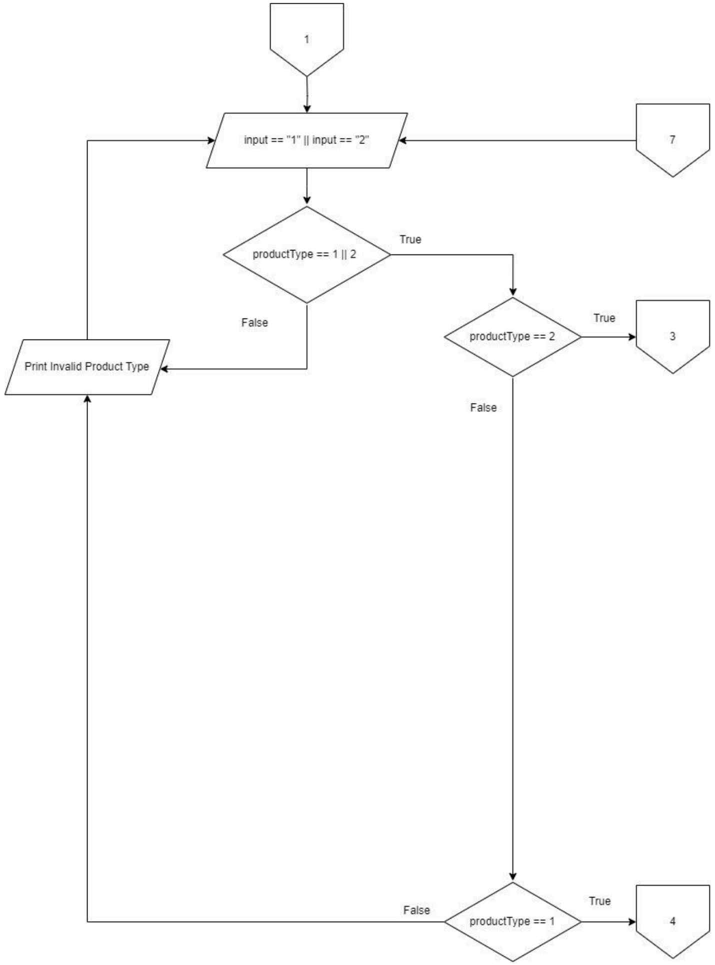
```
If (monthly grower 3-leafy) then
totalPrice=montlyQuantity * 270
                    totalItem=montlyQuantity
END IF


(items)
PRINT “Pick a vegetable produce:
1. Monthly Grower 1 - Leafy, 2. Monthly Grower 2 - Leafy, 3. Monthly Grower 3 - Leafy, 4. Monthly Grower 1 - Fruity, 5. Monthly Grower 2 -
Fruity”
Switch(vegeProduce)
                    Case 1 : “You've picked Monthly Grower 1 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad & Asian Mix
Enter chosen items:”
totalPrice =monthlyQuantity * 105
                                        totalItem =monthlyQuantity,
BREAK
Case 2 : “You've picked Monthly Grower 2 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad & Asian Mix
Enter chosen items:”
totalPrice=monthlyQuantity *189
                                        totalItem = monthlyQuantity,
, BREAK
Case 3 : “You've picked Monthly Grower 3 - Leafy
Pick items:  1. Salad lettuces, 2. Hearty Asian greens, 3. Salad & Asian Mix
Enter chosen items:”
totalPrice=monthlyQuantity * 270
totalItem=monthlyQuantity,
, BREAK
Case 4 : “You've picked Monthly Grower 1 - Fruity
Pick items:  Red/yellow capsicums, tomatoes/cherry, green/red chilis english/cocktail cucumbers
Insert quantity:”
```

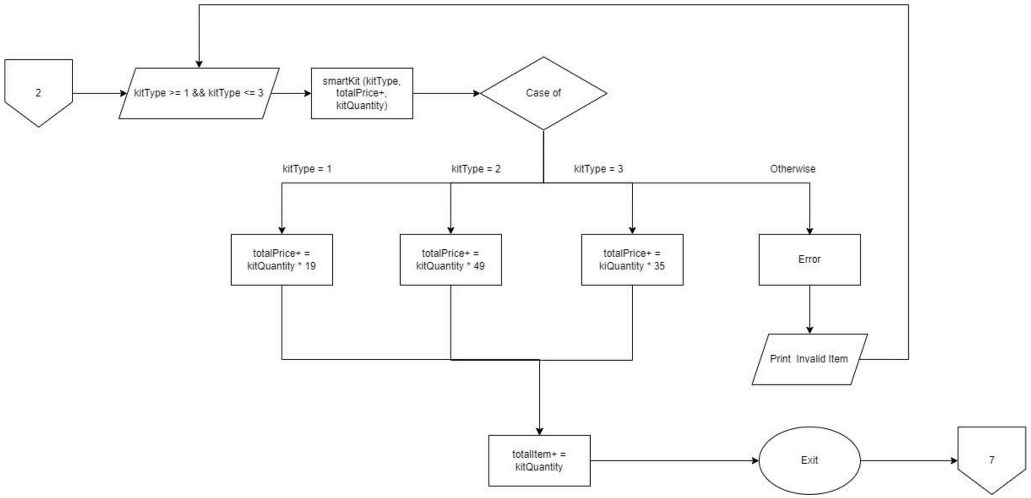
```
totalPrice = monthlyQuantity * 120
totalItem = monthlyQuantity, BREAK
Case 5 : “You've picked Monthly Grower 2 - Fruity
Pick items:  Red/yellow capsicums, tomatoes/cherry, green/red chilis english/cocktail cucumbers
Insert quantity:”
totalPrice = monthlyQuantity * 240
totalItem = monthlyQuantity, BREAK
END CASE


totalPrice=monthlyQuantity * 270
totalItem=monthlyQuantity
if items >= 1 and items <= 3 or itemException = 1
switch(item)
Case 1 = PRINT “Insert quantity:”
totalPrice = monthlyQuantity * 105
totalItem = monthlyQuantity, BREAK
Case 2 = PRINT “Insert quantity:”
totalPrice = monthlyQuantity * 189
totalItem = monthlyQuantity, BREAK
Case 3 = PRINT “Insert quantity:”
totalPrice += monthlyQuantity * 270
totalItem += monthlyQuantity, BREAK
END CASE
  if (items >= 1 and items <= 3)
PRINT “You've picked Annual Grower 1 - Leafy"
 "Pick items: 1. salad lettuces, 2. hearty asian greens, 3. salad & asian mix"
END IF
if (items >= 1 and items <= 3)
switch(items)
Case 1 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
```

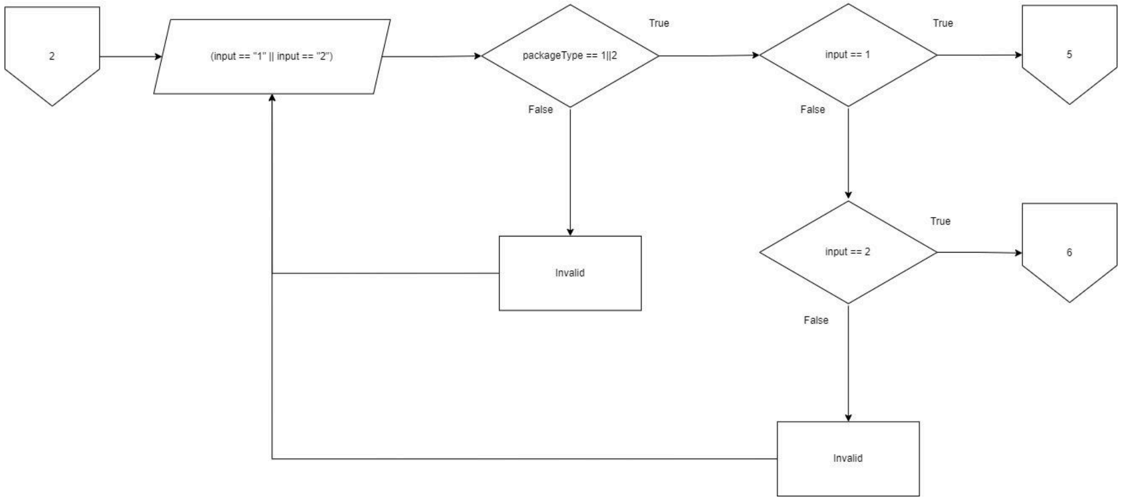
```
totalItem += annualQuantity, BREAK
Case 2 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
totalItem += annualQuantity, BREAK
Case 3 = PRINT “Insert quantity:”
totalPrice += annualQuantity * 1200
totalItem += annualQuantity, BREAK
END CASE
grandTotal(incl.6%) = (totalPrice * 0.06) + totalPrice
totalChange = cash - grandTotal
If totalItems > 0
Display grandTotal
If cash < grandTotal
Output ( "Amount entered is not sufficient. Please enter amount: RM" )
Else
Output ( "Thank you for shopping with us! come again!" )
End if


Display
        Printing receipt
        Salad finger co.
        Date and time
        Location
        Order number
        Invoice
        Bill number
        status
        total quantity
        total price(RM)
        grand total (incl.6%)
        cash
        change
        Thank you , please continue to use our service
END
```

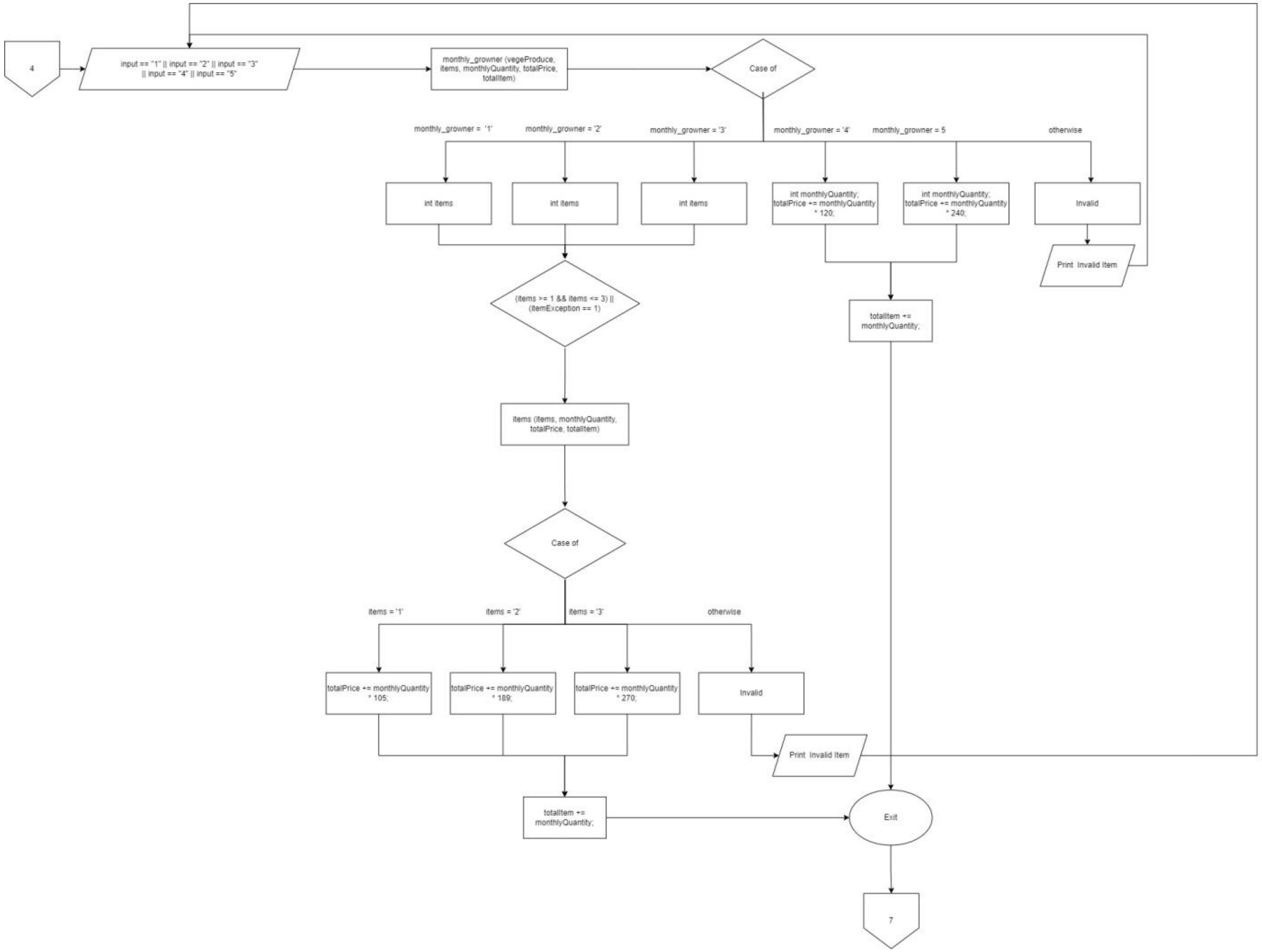
FLOWCHART

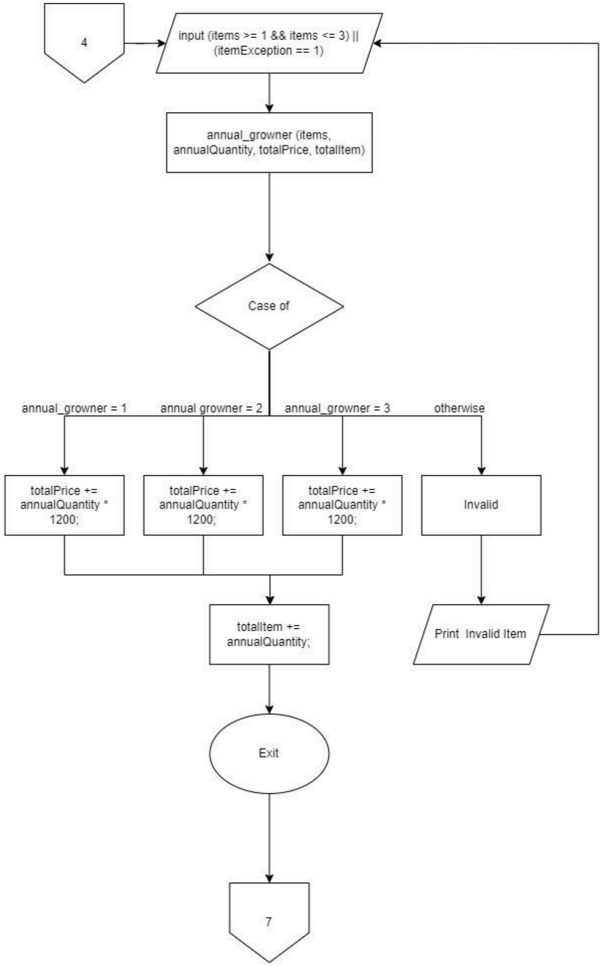

Start
Int productType, packageType, vegeProduce, kitType, itemException;
double totalPrice = 0, grandTotal, cash = 0, totalChange;
int monthlyQuantity, annualQuantity, items = 0, kitQuantity, totalItem = 0, orderNum = 0, billNum = 1234;
char continueBuying, hyphen = 45;
orderNum++;
billNum++;
2

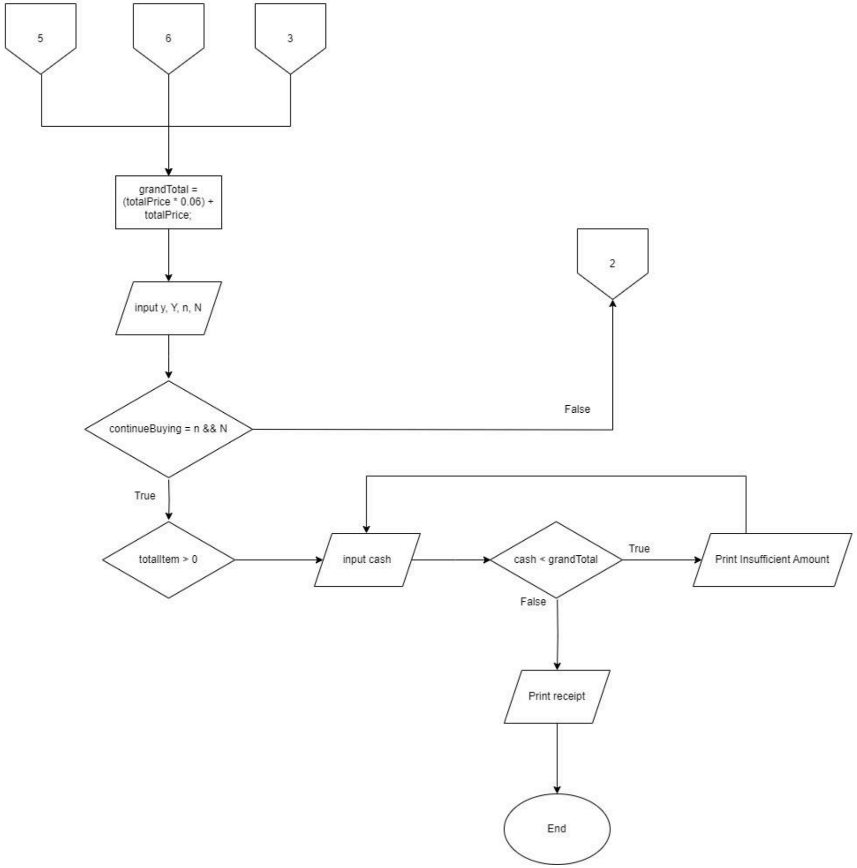

1
input == "1" || input == "2"
7
productType == 1 || 2
True
False
Print Invalid Product Type
productType == 2
True
3
False
False
productType == 1
True
4


2
kitType >= 1 && kitType <= 3
smartKit (kitType, totalPrice+, kitQuantity)
Case of
kitType = 1
kitType = 2
kitType = 3
Otherwise
totalPrice+ = kitQuantity * 19
totalPrice+ = kitQuantity * 49
totalPrice+ = kiQuantity * 35
Error
Print Invalid Item
totalItem+ = kitQuantity
Exit
7


2
(input == "1" || input == "2")
packageType == 1||2
True
False
Invalid
input == 1
True
False
5
input == 2
True
False
6
Invalid


4
input == "1" || input == "2" || input == "3" || input == "4" || input == "5"
monthly_growner (vegeProduce, items, monthlyQuantity, totalPrice, totalItem)
Case of
monthly_growner = '1'
monthly_growner = '2'
monthly_growner = '3'
monthly_growner = '4'
monthly_growner = 5
otherwise
int items
int items
int items
int monthlyQuantity; totalPrice += monthlyQuantity * 120;
int monthlyQuantity; totalPrice += monthlyQuantity * 240;
Invalid
Print Invalid Item
(items >= 1 && items <= 3) || (itemException == 1)
totalItem += monthlyQuantity;
items (items, monthlyQuantity, totalPrice, totalItem)
Case of
items = '1'
items = '2'
items = '3'
otherwise
totalPrice += monthlyQuantity * 105;
totalPrice += monthlyQuantity * 189;
totalPrice += monthlyQuantity * 270;
Invalid
Print Invalid Item
totalItem += monthlyQuantity;
Exit
7


4
input (items >= 1 && items <= 3) || (itemException == 1)
annual_growner (items, annualQuantity, totalPrice, totalItem)
Case of
annual_growner = 1
annual growner = 2
annual_growner = 3
otherwise
totalPrice += annualQuantity * 1200;
totalPrice += annualQuantity * 1200;
totalPrice += annualQuantity * 1200;
Invalid
totalItem += annualQuantity;
Print Invalid Item
Exit
7


5
6
3
grandTotal = (totalPrice * 0.06) + totalPrice;
2
input y, Y, n, N
False
continueBuying = n && N
True
totalItem > 0
input cash
cash < grandTotal
True
Print Insufficient Amount
False
Print receipt
End

C++ SOURCE CODE

```
#include <iostream>
#include <ctime>//time&date for receipt area
#include <limits>//the limit header for input validation, avoid infinity looping
using namespace std;
int main()
{
int productType, packageType, vegeProduce, kitType, itemException;
double totalPrice = 0, grandTotal, cash = 0, totalChange;     // Initialize totalPrice to 0
int monthlyQuantity, annualQuantity, items = 0, kitQuantity, totalItem = 0, orderNum = 0, billNum = 1234;       // Initialize totalItem to 0
char continueBuying, hyphen = 45;   // ASCII value for '-';
orderNum++;
billNum++;

do
{
    //validation loop for productType, avoid infinite loop
    while (true)
    {
       cout << "\nPick a product:\n1. vegetable subscription\n2. smart kit" << endl << "\nEnter chosen product: ";
       string input;
       cin >> input;

       if (input == "1" || input == "2")
       {
          productType = stoi(input); // Convert the string to an integer
          break;
       }
       else
       {
          cout << "Invalid product, pick again!" << endl;
```

```
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
            }
        }
if (productType == 2)
{
            while (true) {
                cout << "\nPick a package type:\n1. farm kit\n2. compost kit\n3. microgreens kit" << endl << "\nEnter chosen package: ";
                string input;
                cin >> input;
                if (input == "1" || input == "2" || input == "3") {
                    kitType = stoi(input);
                    break;
                } else {
                    cout << "Invalid item, pick again!" << endl;
                    cin.clear();
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                }
            }

/*          cout <<             "\nPick a package type:\n1. farm kit\n2. compost kit\n3. microgreens kit" << endl << "\nEnter chosen package: ";
cin >> kitType;*/

if (kitType >= 1 && kitType <= 3)
{
switch (kitType)
{
case 1:
cout << "You've picked Farm Kit" << endl <<
"\nInsert quantity: ";
while (!(cin >> kitQuantity)) {
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += kitQuantity * 19;            // Add to total price
totalItem += kitQuantity;        // Add to total items
cout << "\nTotal price: RM" << totalPrice << endl;
break;
case 2:
cout << "You've picked Compost Kit" << endl <<
"\nInsert quantity: ";
while (!(cin >> kitQuantity)) {
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += kitQuantity * 49;
totalItem += kitQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
break;
case 3:
cout << "You've picked Microgreens Kit" << endl <<
"\nInsert quantity: ";
while (!(cin >> kitQuantity)) {
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += kitQuantity * 35;
totalItem += kitQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
break;
}
```

```
}
else
cout << "\nInvalid kit, pick again!";
}
else if (productType == 1)
{

   while (true)
         {
            cout << "\nPick a package type:\n1. monthly grower\n2. annual grower\nEnter chosen package: ";
            string input;
            cin >> input;

            if (input == "1" || input == "2")
            {
            packageType = stoi(input); // Convert the string to an integer
            break;
            }
            else
            {
            cout << "Invalid product, pick again!" << endl;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            }
         }
/*cout <<"\nPick a package type:\n1. monthly grower\n2. annual grower\nEnter chosen package: ";
cin >> packageType;*/

if (packageType == 1)
{

   while (true)
```

```
                {
                   cout <<"\nPick a vegetable produce:\n1. Monthly Grower 1 - Leafy\n2. Monthly Grower 2 - Leafy\n3. Monthly Grower 3 -
Leafy\n4. Monthly Grower 1 - Fruity\n5. Monthly Grower 2 - Fruity\n";
   cout << "\nEnter a chosen vegetable produce: ";
                   string input;
                   cin >> input;

                   if (input == "1" || input == "2" || input == "3" || input == "4" || input == "5")
                   {
                   vegeProduce = stoi(input); // Convert the string to an integer
                   break;
                   }
                   else
                   {
                   cout << "Invalid product, pick again!" << endl;
                   cin.clear();
                   cin.ignore(numeric_limits<streamsize>::max(), '\n');
                   }
                }

/*cout <<"\nPick a vegetable produce:\n1. Monthly Grower 1 - Leafy\n2. Monthly Grower 2 - Leafy\n3. Monthly Grower 3 - Leafy\n4. Monthly
Grower 1 - Fruity\n5. Monthly Grower 2 - Fruity\n";
cout << "\nEnter a chosen vegetable produce: ";
cin >> vegeProduce;*/

if (vegeProduce >= 1 && vegeProduce <= 5)
{
switch (vegeProduce)
{
case 1:
cout <<"\nYou've picked Monthly Grower 1 - Leafy \nPick items: \n1. Salad lettuces\n2. Hearty Asian greens\n3. Salad & Asian Mix"<< endl;
cout << "\nEnter chosen items: ";
```

```
cin >> items;

break;
case 2:
cout <<          "\nYou've picked Monthly Grower 2 - Leafy \nPick items: \n1. Salad lettuces\n2. Hearty Asian greens\n3. Salad & Asian Mix" <<
endl;
cout << "\nEnter chosen items: ";
cin >> items;

break;
case 3:
cout <<          "\nYou've picked Monthly Grower 3 - Leafy \nPick items: \n1. Salad lettuces\n2. Hearty Asian greens\n3. Salad & Asian Mix" <<
endl;
cout << "\nEnter chosen items: ";
cin >> items;

break;
case 4:
cout <<          "\nYou've picked Monthly Grower 1 - Fruity\nThis package includes: \n-red/yellow capsicums\n-tomatoes/cherry\n-green/red
chilis\n-english/cocktail cucumbers"<< endl << "\nInsert quantity: ";
while (!(cin >> monthlyQuantity)) {
                 cout << "Invalid input. \nPlease enter a numeric quantity: ";
                 cin.clear();
                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
                 }
totalPrice += monthlyQuantity * 120;
totalItem += monthlyQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
itemException = 1;
break;
case 5:
```

```
cout <<"\nYou've picked Monthly Grower 2 - Fruity\nThis package includes: \n-red/yellow capsicums\n-tomatoes/cherry\n-green/red chilis\n-
english/cocktail cucumbers"   << endl << "\nInsert quantity: ";
while (!(cin >> monthlyQuantity)) {
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                        }
totalPrice += monthlyQuantity * 240;
totalItem += monthlyQuantity;
cout << "\nTotal price: " << totalPrice << endl;
itemException = 1;
break;
}
if ((items >= 1 && items <= 3) || (itemException == 1))
{
switch (items)
{
case 1:
cout << "\nInsert quantity: ";
while (!(cin >> monthlyQuantity)) {
                            cout << "Invalid input. \nPlease enter a numeric quantity: ";
                            cin.clear();
                            cin.ignore(numeric_limits<streamsize>::max(), '\n');
                            }
totalPrice += monthlyQuantity * 105;
totalItem += monthlyQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
case 2:
cout << "\nInsert quantity: ";
while (!(cin >> monthlyQuantity)) {
```

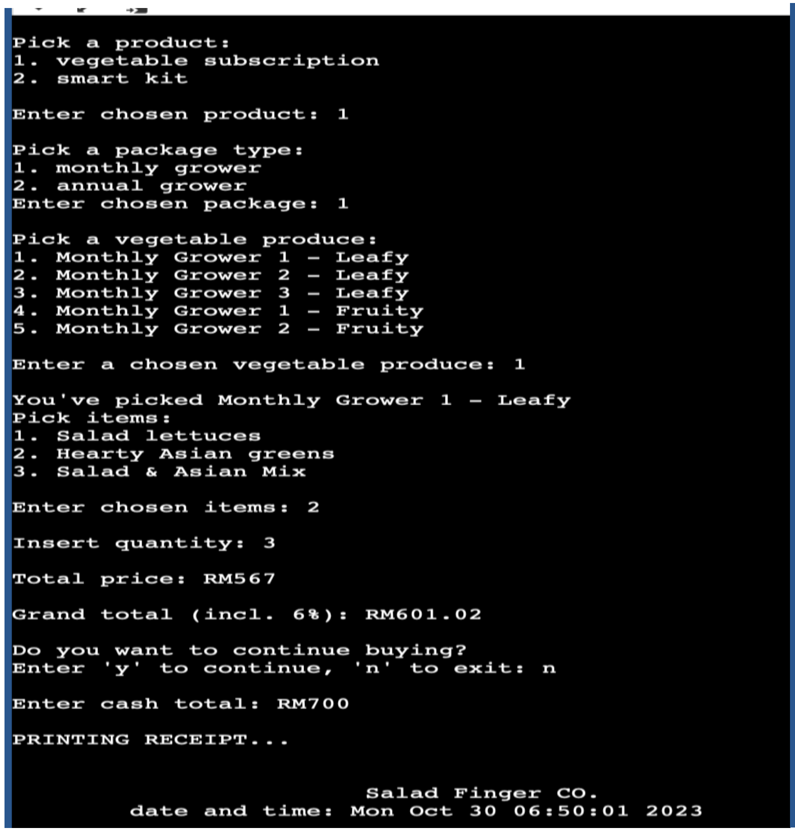
```
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                        }
totalPrice += monthlyQuantity * 189;
totalItem += monthlyQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
case 3:
cout << "\nInsert quantity: ";
while (!(cin >> monthlyQuantity)) {
                        cout << "Invalid input. \nPlease enter a numeric quantity: ";
                        cin.clear();
                        cin.ignore(numeric_limits<streamsize>::max(), '\n');
                        }
totalPrice += monthlyQuantity * 270;
totalItem += monthlyQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
}
}
else
cout << "\nInvalid item, pick again!";
}
else
cout << "\nInvalid item, pick again!";
}
else if (packageType == 2)
{
   while (true)
           {
             cout << "You've picked Monthly Grower 1 - Leafy" << endl;
             cout << "Pick items:" << endl;
             cout << "1. Salad lettuces" << endl;
             cout << "2. Hearty Asian greens" << endl;
             cout << "3. Salad & Asian Mix" << endl;
             cout << "Enter chosen items: ";

             if (cin >> items)
             {
                if (items >= 1 && items <= 3)
                {
                break; // Valid input, exit the loop
                }
             }
             else
             {
             cout << "Invalid item, pick again!" << endl;
             cin.clear();
             cin.ignore(numeric_limits<streamsize>::max(), '\n');
             }
           }


/*cout << "\nYou've picked Annual Grower 1 - Leafy" << endl <<"\nPick items:\n1. salad lettuces\n2. hearty asian greens\n3. salad & asian
mix\n";
cout << "\nEnter chosen items: ";
cin >> items;*/

if (items >= 1 && items <= 3)
{
```

```
switch (items)
{
case 1:
cout << "\nInsert quantity: ";
while (!(cin >> annualQuantity)) {
                    cout << "Invalid input. \nPlease enter a numeric quantity: ";
                    cin.clear();
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += annualQuantity * 1200;
totalItem += annualQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
case 2:
cout << "\nInsert quantity: ";
while (!(cin >> annualQuantity)) {
                    cout << "Invalid input. \nPlease enter a numeric quantity: ";
                    cin.clear();
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += annualQuantity * 1200;
totalItem += annualQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
case 3:
cout << "\nInsert quantity: ";
while (!(cin >> annualQuantity)) {
                    cout << "Invalid input. \nPlease enter a numeric quantity: ";
                    cin.clear();
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    }
totalPrice += annualQuantity * 1200;
totalItem += annualQuantity;
cout << "\nTotal price: RM" << totalPrice << endl;
items = 0;
break;
}
}
else
cout << "\nInvalid item, pick again!";
}
else
{
cout << "\nInvalid product, pick again!";
}
}
else
{
cout << "\nInvalid product, pick again!";
}
grandTotal = (totalPrice * 0.06) + totalPrice;
cout << "\nGrand total (incl. 6%): RM" << grandTotal << "\n";
cout << "\nDo you want to continue buying?\nEnter 'y' to continue, 'n' to exit: ";
cin >> continueBuying;

} while (continueBuying != 'n' && continueBuying != 'N'); // Continue buying as long as 'n' or 'N' is not entered

if (totalItem > 0)
{
cout << "\nEnter cash total: RM";
cin >> cash;
while (cash < grandTotal) {
```

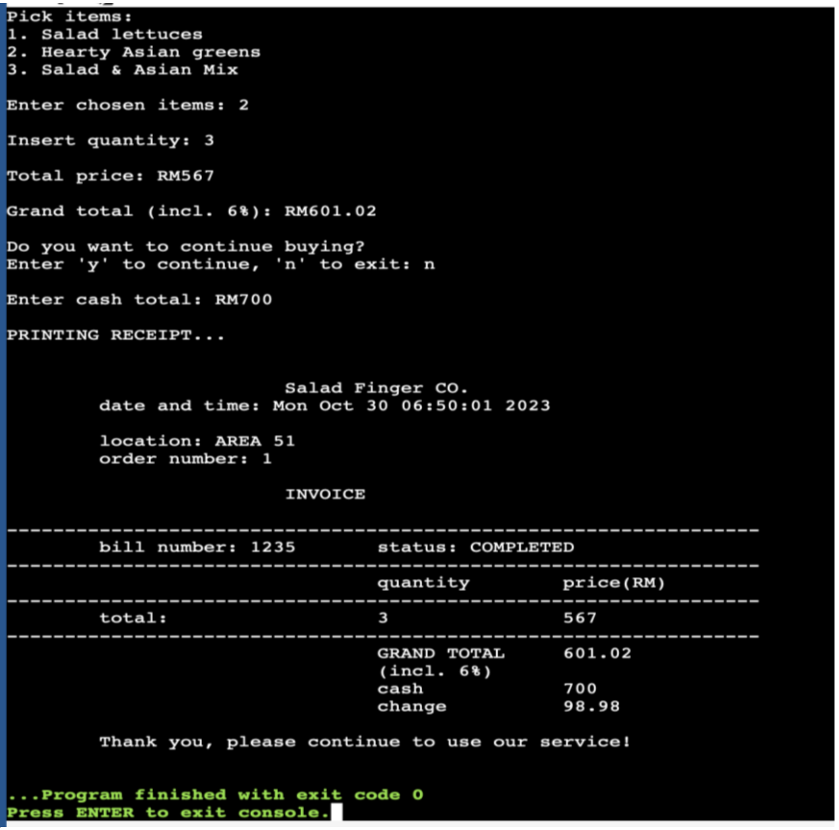
```
cout << "Amount entered is not sufficient\nPlease enter amount: RM";
cin >> cash;
}
cout << "\nPRINTING RECEIPT..." << endl;
//receipt area
cout << "\n\n\t\t\tSalad Finger CO." << endl;
time_t now = time(0);
char* date_time = ctime(&now);
cout << "\tdate and time: " << date_time << endl;
cout << "\tlocation: AREA 51" << endl;
cout << "\torder number: " << orderNum << endl;
cout << "\n\t\t\tINVOICE\n" << endl;
int hyphenCount = 65; //print 65 '-'
for (int i = 0; i < hyphenCount; i++) {
cout << '-';
}
cout << endl;
cout << "\tbill number: " << billNum << "\tstatus: COMPLETED" << endl;
//print 65 '-'
for (int i = 0; i < hyphenCount; i++) {
cout << '-';
}
cout << endl;
cout << "\t\t\t\tquantity\tprice(RM)" << endl;
//print 65 '-'
for (int i = 0; i < hyphenCount; i++) {
cout << '-';
}
cout << endl;
cout << "\ttotal:\t\t" << "\t" << totalItem << "\t\t" << totalPrice << endl;
//print 65 '-'
for (int i = 0; i < hyphenCount; i++) {
```

```
cout << '-';
}
cout << endl;
cout << "\t\t\t\tGRAND TOTAL\t" << grandTotal << "\n\t\t\t\t(incl. 6%)"<<endl;
cout << "\t\t\t\tcash\t\t" << cash << endl;
totalChange = cash - grandTotal;
cout << "\t\t\t\tchange\t\t" << totalChange << endl;
cout << "\n\tThank you, please continue to use our service!" << endl;

}
else
{
cout << "\nThank you for shopping with us! come again!";
}

return 0;
}
```

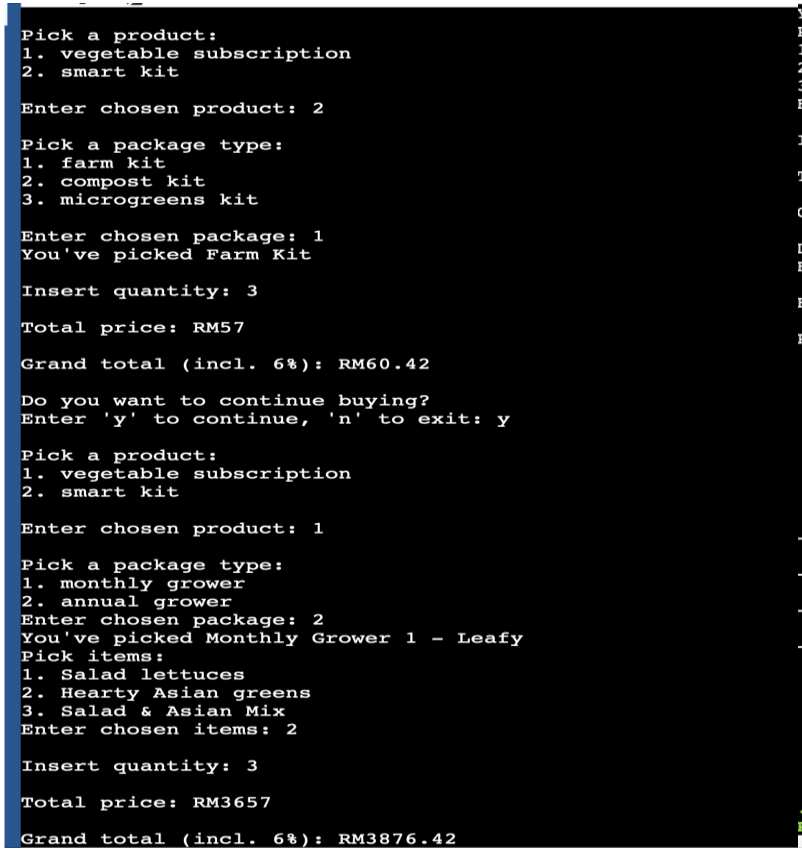
SAMPLE OUTPUT



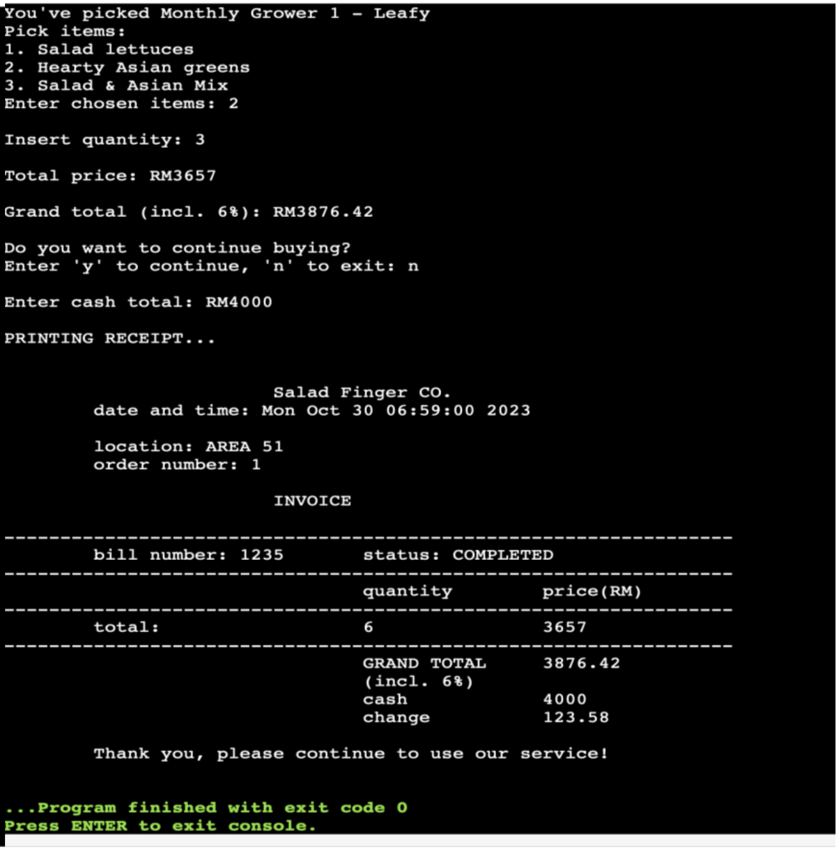
```
Pick a product:
1. vegetable subscription
2. smart kit

Enter chosen product: 1

Pick a package type:
1. monthly grower
2. annual grower
Enter chosen package: 1

Pick a vegetable produce:
1. Monthly Grower 1 – Leafy
2. Monthly Grower 2 – Leafy
3. Monthly Grower 3 – Leafy
4. Monthly Grower 1 – Fruity
5. Monthly Grower 2 – Fruity

Enter a chosen vegetable produce: 1

You've picked Monthly Grower 1 – Leafy
Pick items:
1. Salad lettuces
2. Hearty Asian greens
3. Salad & Asian Mix

Enter chosen items: 2

Insert quantity: 3

Total price: RM567

Grand total (incl. 6%): RM601.02

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: n

Enter cash total: RM700

PRINTING RECEIPT...


                        Salad Finger CO.
        date and time: Mon Oct 30 06:50:01 2023
```



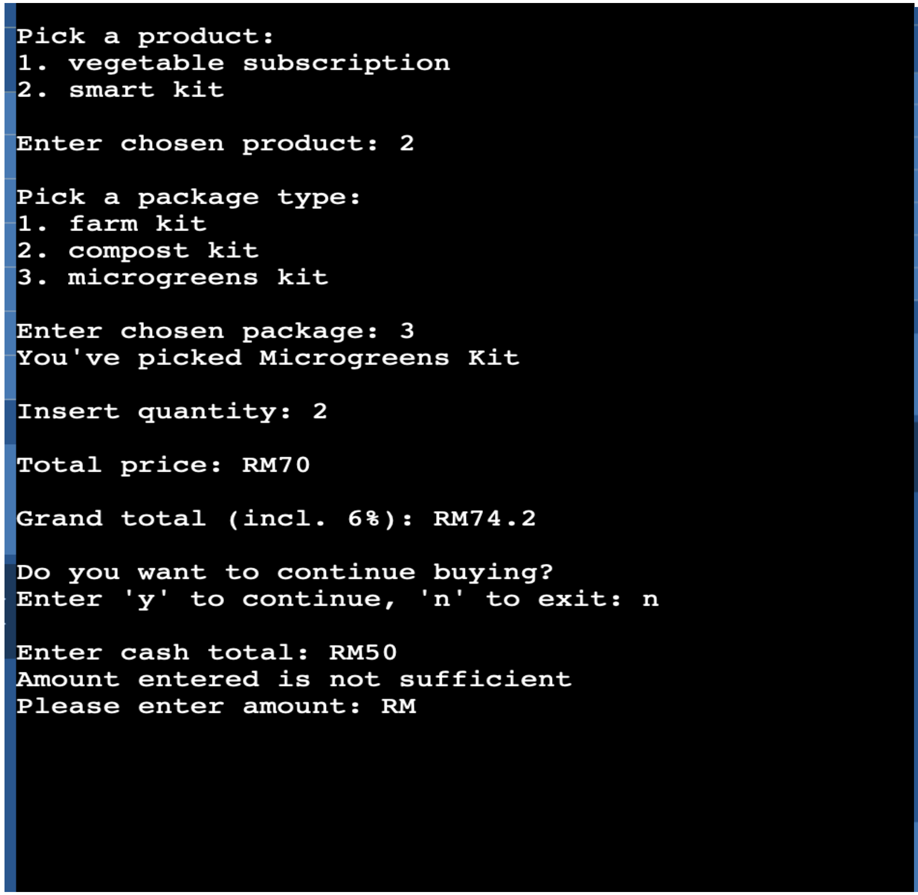
```
Pick items:
1. Salad lettuces
2. Hearty Asian greens
3. Salad & Asian Mix

Enter chosen items: 2

Insert quantity: 3

Total price: RM567

Grand total (incl. 6%): RM601.02

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: n

Enter cash total: RM700

PRINTING RECEIPT...


                        Salad Finger CO.
        date and time: Mon Oct 30 06:50:01 2023

        location: AREA 51
        order number: 1

                        INVOICE

-----------------------------------------------------------------
        bill number: 1235       status: COMPLETED
-----------------------------------------------------------------
                                quantity        price(RM)
-----------------------------------------------------------------
        total:                  3               567
-----------------------------------------------------------------
                                GRAND TOTAL     601.02
                                (incl. 6%)
                                cash            700
                                change          98.98

        Thank you, please continue to use our service!


...Program finished with exit code 0
Press ENTER to exit console.
```



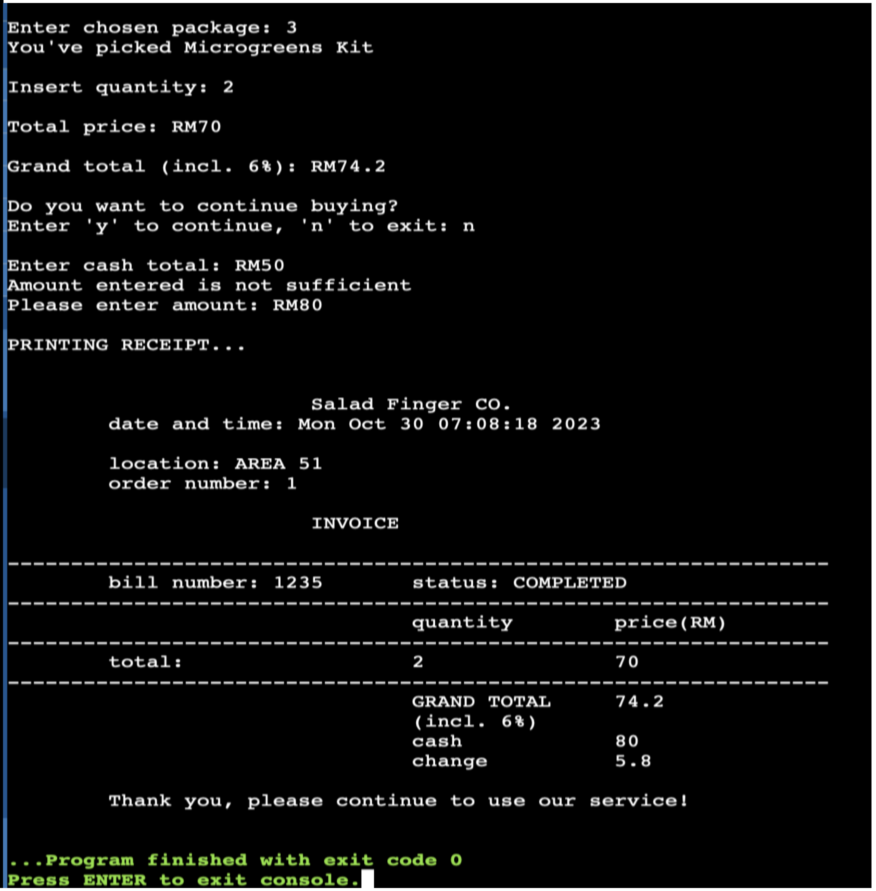
```
Pick a product:
1. vegetable subscription
2. smart kit

Enter chosen product: 2

Pick a package type:
1. farm kit
2. compost kit
3. microgreens kit

Enter chosen package: 1
You've picked Farm Kit

Insert quantity: 3

Total price: RM57

Grand total (incl. 6%): RM60.42

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: y

Pick a product:
1. vegetable subscription
2. smart kit

Enter chosen product: 1

Pick a package type:
1. monthly grower
2. annual grower
Enter chosen package: 2
You've picked Monthly Grower 1 - Leafy
Pick items:
1. Salad lettuces
2. Hearty Asian greens
3. Salad & Asian Mix
Enter chosen items: 2

Insert quantity: 3

Total price: RM3657

Grand total (incl. 6%): RM3876.42
```



```
You've picked Monthly Grower 1 - Leafy
Pick items:
1. Salad lettuces
2. Hearty Asian greens
3. Salad & Asian Mix
Enter chosen items: 2

Insert quantity: 3

Total price: RM3657

Grand total (incl. 6%): RM3876.42

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: n

Enter cash total: RM4000

PRINTING RECEIPT...


                                Salad Finger CO.
          date and time: Mon Oct 30 06:59:00 2023

          location: AREA 51
          order number: 1

                                INVOICE

-----------------------------------------------------------------
          bill number: 1235         status: COMPLETED
-----------------------------------------------------------------
                                    quantity         price(RM)
-----------------------------------------------------------------
          total:                    6                3657
-----------------------------------------------------------------
                                    GRAND TOTAL      3876.42
                                    (incl. 6%)
                                    cash             4000
                                    change           123.58

          Thank you, please continue to use our service!


...Program finished with exit code 0
Press ENTER to exit console.
```



```
Pick a product:
1. vegetable subscription
2. smart kit

Enter chosen product: 2

Pick a package type:
1. farm kit
2. compost kit
3. microgreens kit

Enter chosen package: 3
You've picked Microgreens Kit

Insert quantity: 2

Total price: RM70

Grand total (incl. 6%): RM74.2

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: n

Enter cash total: RM50
Amount entered is not sufficient
Please enter amount: RM
```



```
Enter chosen package: 3
You've picked Microgreens Kit

Insert quantity: 2

Total price: RM70

Grand total (incl. 6%): RM74.2

Do you want to continue buying?
Enter 'y' to continue, 'n' to exit: n

Enter cash total: RM50
Amount entered is not sufficient
Please enter amount: RM80

PRINTING RECEIPT...


                        Salad Finger CO.
        date and time: Mon Oct 30 07:08:18 2023

        location: AREA 51
        order number: 1

                        INVOICE

-----------------------------------------------------------------
        bill number: 1235       status: COMPLETED
-----------------------------------------------------------------
                                quantity        price(RM)
-----------------------------------------------------------------
        total:                  2               70
-----------------------------------------------------------------
                                GRAND TOTAL     74.2
                                (incl. 6%)
                                cash            80
                                change          5.8

        Thank you, please continue to use our service!


...Program finished with exit code 0
Press ENTER to exit console.
```