**Questions for Task B1**

A Satay Restaurant recently launched an online ordering system for its food delivery service. Customers can choose from three combo menu as follows:

|  |  |  |
| --- | --- | --- |
| Combo Type | Item | Price (RM) |
| A  B C | 25 Chicken Satay + 25 Beef Satay  30 Chicken Satay + 20 Mutton Satay  10 Mutton Satay + 40 Beef Satay | 40.00  52.00  46.00 |

Ketupat is also available at RM 0.60 each.

Construct an algorithm that read the combo type chosen by the customer, quantity, add-on option and the quantity of Ketupat if the customer wishes to order, then calculate and display the order price.

**FlowChart for Task B1**

Prompt the user to enter combo type

Display Selection List

Start

Declare variable comboType, Qty, price, ordPrice, addOn, addQty and addPrice.

A

Input Qty

Prompt the user to enter food quantity

comboType=toupper(comboType)

Input comboType

B

Prompt the user to enter adds-on

Input addOn

addOn=toupper(addOn)

ordPrice=price\*Qty

False

False

False

True

True

True

price=0.00

price=52.00

price=46.00

price=40.00

comboType==’B’

?

comboType==’C’

?

comboType==’A’

?

A

ordPrice=ordPrice+addQty\*addPrice

End

Input addQty

Prompt the user to enter adds-on quantity

B

False

addOn==’Y’

?

True

addPrice=0.60

addPrice=0.00

Display ordPrice

**Source Code for Task B1**

**/\*Name:Ng Ke Rong\*/**

**/\*I/C No:990124-07-5449\*/**

**/\*Date:21June2018\*/**

**#include <stdio.h>**

**int main()**

**{**

**/\*Declaration\*/**

**char comboType;**

**int Qty;**

**char addOn;**

**int addQty;**

**float ordPrice;**

**float price;**

**float addPrice;**

**/\*Display Selection List\*/**

**clrscr();**

**printf("\t\t\t\tSATAY RESTAURANT\n");**

**printf("\t-----------------------------------------------------------------\n");**

**printf("\tCombo Type\t\t\tItem\t\t\tPrice(RM)\n");**

**printf("\t-----------------------------------------------------------------\n");**

**printf("\t A\t\t25 Chicken Satay+25Beef Satay\t\t40.00\n");**

**printf("\t B\t\t30 Chicken Satay+20 Mutton Satay\t52.00\n");**

**printf("\t C\t\t10 Mutton Satay+40 Beef Satay\t\t46.00\n");**

**printf("\tAdd-On\t\t1 Ketupat\t\t\t\t0.60\n");**

**printf("\t-----------------------------------------------------------------\n\n");**

**/\*Input Combo Type\*/**

**printf("\tEnter Combo Type(A/B/C)\t:");**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**/\*Input Quantity\*/**

**printf("\tEnter quantity\t\t:");**

**scanf("%d",&Qty);**

**fflush(stdin);**

**/\*Selection\*/**

**if(comboType=='A')**

**price=40.00;**

**else if(comboType=='B')**

**price=52.00;**

**else if(comboType=='C')**

**price=46.00;**

**else**

**price=0.00;**

**/\*Calculate Order Price\*/**

**ordPrice=price\*Qty;**

**/\*Input Add-On\*/**

**printf("\tAdd-on Ketupat(Y/N)\t:");**

**scanf("%c",&addOn);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**addOn=toupper(addOn);**

**/\*Input Add-On Quantity\*/**

**printf("\tEnter Ketupat quantity\t:");**

**scanf("%d",&addQty);**

**/\*Selection\*/**

**if(addOn=='Y')**

**addPrice=0.60;**

**else**

**addPrice=0.00;**

**/\*Calculate Order Price\*/**

**ordPrice+=addQty\*addPrice;**

**/\*Display Order Price\*/**

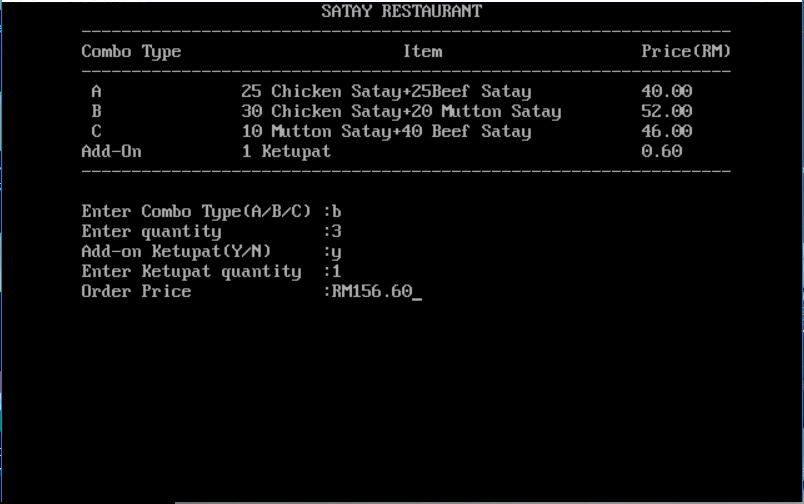
**printf("\tOrder Price\t\t:RM%.2f",ordPrice);**

**getch ();**

**return 0;**

**}**

**Input and Output for Task B1**

* Sample Input and Output with ***correct*** combo type
* Sample Input and Output with ***wrong*** combo type

Questions for Task B2

Based on, construct an algorithm that allow customer to order combo type, calculate, display the total order price has to pay.

**FlowChart for Task B2**

C

D

A

True

False

comboType!=’X’

?

comboType=toupper(comboType)

Input comboType

Prompt the user to enter first combo type

Display Selection List

totalPrice=0

counter=1

Declare variable comboType, Qty, price, ordPrice, addOn, addQty, addPrice, counter, addOrd and totalPrice.

Start

A

price=0.00

ordPrice=price\*Qty

Prompt the user to enter food quantity

Input Qty

comboType==’A’

?

False

False

False

comboType==’C’

?

comboType==’B’

?

B

Display ordPrice

price=52.00

True

price=46.00

True

True

price=40.00

totalPrice=totalPrice+ordPrice

counter=counter+1

Input comboType

Prompt the user to enter next combo type

C

comboType=toupper(comboType)

B

D

addOn=toupper(addOn)

Input addOn

Prompt the user to enter adds-on

addOrd=addQty\*addPrice

addPrice=0.00

addOn==’Y’

?

Input addQty

Prompt the user to enter adds-on quantity

Display addOrd

E

False

True

addPrice=0.60

End

Display totalPrice

totalPrice=totalPrice+addOrd

E

**Source Code for Task B2**

**/\*Name:Ng Ke Rong\*/**

**/\*I/C No:990124-07-5449\*/**

**/\*Date:21June2018\*/**

**#include <stdio.h>**

**int main()**

**{**

**/\*Declaration\*/**

**char comboType;**

**int Qty;**

**char addOn;**

**int addQty;**

**float ordPrice;**

**float price;**

**float addPrice;**

**int counter;**

**float totalPrice;**

**int Line;**

**float addOrd;**

**/\*Initialize\*/**

**counter=1;**

**totalPrice=0;**

**/\*Display Selection List\*/**

**clrscr();**

**printf("\t\t\t\tSATAY RESTAURANT\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tCombo Type\t\t\tItem\t\t\tPrice(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\t A\t\t25 Chicken Satay+25Beef Satay\t\t40.00\n");**

**printf("\t B\t\t30 Chicken Satay+20 Mutton Satay\t52.00\n");**

**printf("\t C\t\t10 Mutton Satay+40 Beef Satay\t\t46.00\n");**

**printf("\tAdd-On\t\t1 Ketupat\t\t\t\t0.60\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Input Combo Type\*/**

**printf("\n\n\t%d. Enter Combo Type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**/\*Loop\*/**

**while(comboType!='X')**

**{**

**/\*Input Quantity\*/**

**printf("\tEnter quantity\t\t\t\t:");**

**scanf("%d",&Qty);**

**fflush(stdin);**

**/\*Selection\*/**

**if(comboType=='A')**

**price=40.00;**

**else if(comboType=='B')**

**price=52.00;**

**else if(comboType=='C')**

**price=46.00;**

**else**

**price=0.00;**

**/\*Calculate Order Price\*/**

**ordPrice=price\*Qty;**

**/\*Display Order Price\*/**

**printf("\tOrder Price for combo %c\t\t\t:RM%.2f\n\n",comboType,ordPrice);**

**/\*Accumulate Total Order Price\*/**

**totalPrice+=ordPrice;**

**/\*Increase counter by one\*/**

**counter++;**

**/\*Input Next Combo Type\*/**

**printf("\t%d. Enter combo type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**}**

**/\*Input Add-On\*/**

**printf("\n\tAdd-on Ketupat(Y/N)\t\t\t:");**

**scanf("%c",&addOn);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**addOn=toupper(addOn);**

**/\*Input Add-On Quantity\*/**

**printf("\tEnter Ketupat quantity\t\t\t:");**

**scanf("%d",&addQty);**

**/\*Selection\*/**

**if(addOn=='Y')**

**addPrice=0.60;**

**else**

**addPrice=0.00;**

**/\*Calculate Order Price\*/**

**addOrd=addPrice\*addQty;**

**/\*Display Add-On\*/**

**printf("\tOrder Price for ketupat\t\t\t:Rm%.2f\n\n",addOrd);**

**/\*Calculate Total Order Price\*/**

**totalPrice+=addOrd;**

**/\*Display Total Order Price\*/**

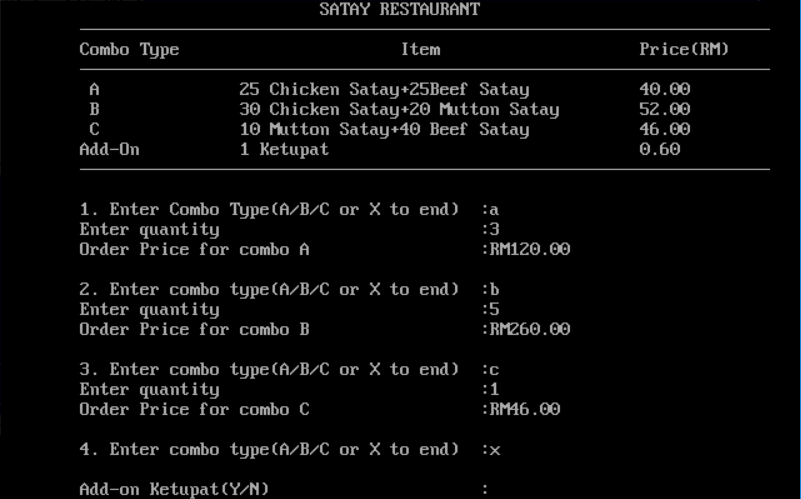
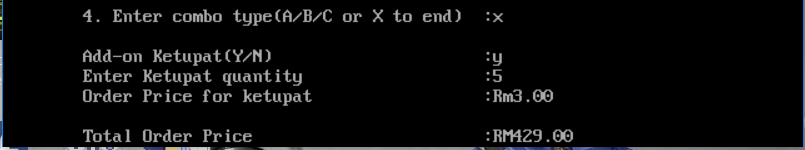
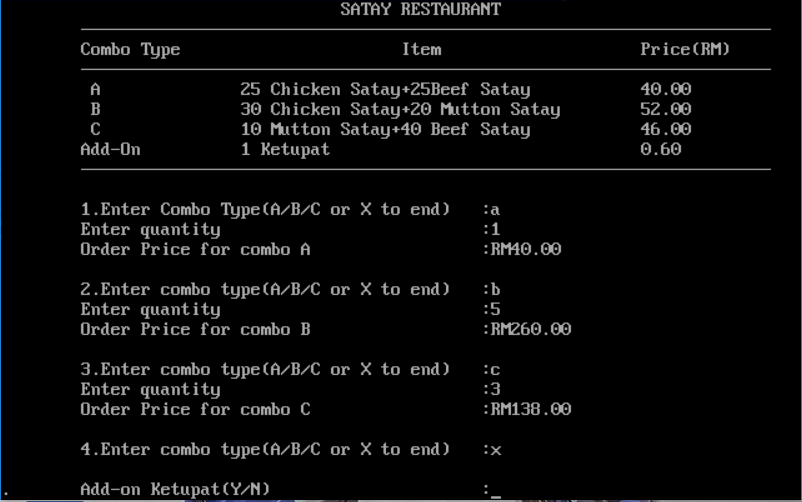
**printf("\tTotal Order Price\t\t\t:RM%.2f",totalPrice);**

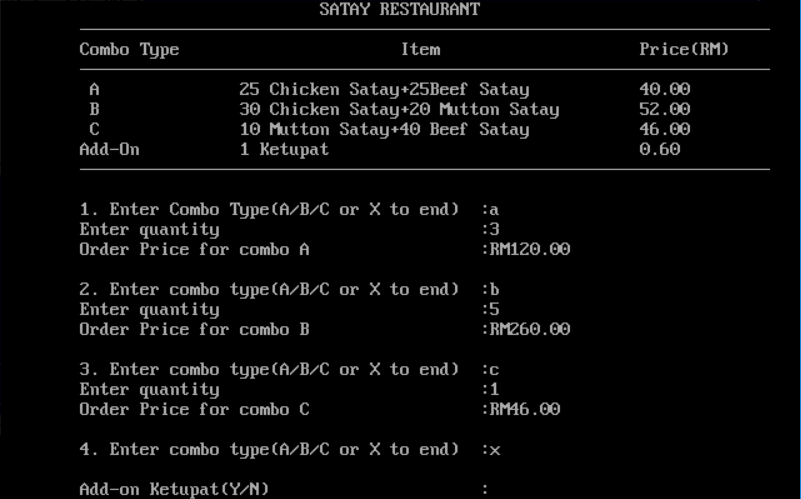
**getch ();**

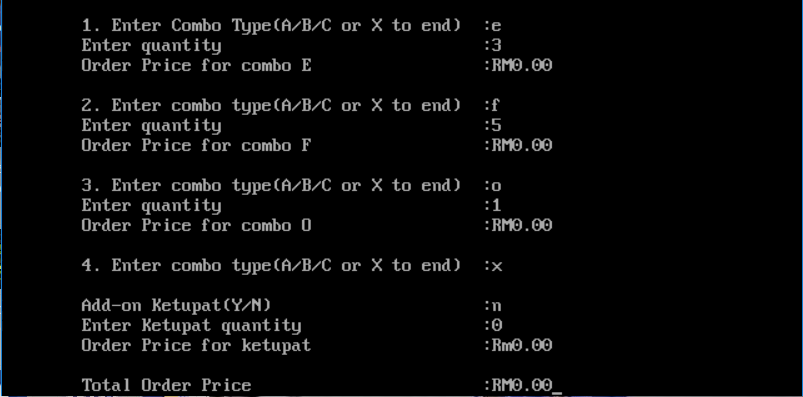
**return 0;**

**}**

**Input and Output for Task B2**

* Sample Input and Output with ***correct*** combo type
* Sample Input and Output with ***wrong*** combo type





Questions for Task B3

Based on Assignment B2, construct an algorithm that implements a function calculatePrice (comboType) that accept combo type.

Construct an algorithm for the main function that read the combo type, call function calculatePrice (comboType), read Ketupat Adds-On option, quantity of Ketupat if any and display the total order price has to pay

**FlowChart for Task B3**

Start

C

False

B

True

Declare variable comboType, Qty, price, ordPrice, addOn, addQty, addPrice, counter, addOrd and totalPrice.

A

comboType!=’X’

?

totalPrice=0

counter=1

comboType=toupper(comboType)

Input comboType

Prompt the user to enter first combo type

Display Selection List

Declare function prototype calculatePrice

B

comboType=toupper(comboType)

Input comboType

Prompt the user to enter next combo type

counter=counter+1

totalPrice=totalPrice+ordPrice

Display ordPrice

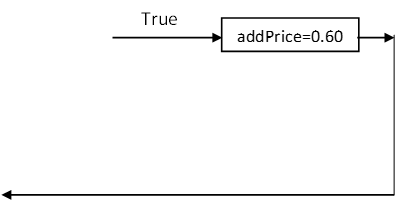
ordPrice=price\*Qty

Call Function

Prompt the user to enter food quantity

Input Qty

A

****

False

D

C

Prompt the user to enter adds-on

Input addOn

addOn=toupper(addOn)

Prompt the user to enter adds-on quantity

Input addQty

addOn==’Y’

?

addPrice=0.00

addOrd=addQty\*addPrice

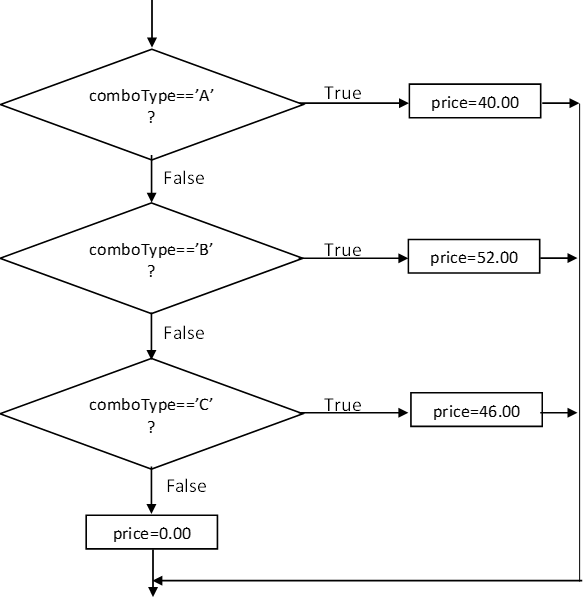
Display addOrd

D

End

Display totalPrice

totalPrice=totalPrice+addOrd

****

return price

Declare variable price

Start calculatePrice

**Source Code for Task B3**

**/\*Name:Ng Ke Rong\*/**

**/\*I/C No:990124-07-5449\*/**

**/\*Date:21June2018\*/**

**#include <stdio.h>**

**int main()**

**{**

**/\*Function Prototype\*/**

**float calculatePrice(char comboType);**

**/\*Declaration\*/**

**char comboType;**

**int Qty;**

**char addOn;**

**int addQty;**

**float ordPrice;**

**float price;**

**float addPrice;**

**int counter;**

**float totalPrice;**

**int Line;**

**float addOrd;**

**/\*Initialize\*/**

**counter=1;**

**totalPrice=0;**

**/\*Display Selection List\*/**

**clrscr();**

**printf("\t\t\t\t SATAY RESTAURANT\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tCombo Type\t\t\tItem\t\t\tPrice(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\t A\t\t25 Chicken Satay+25Beef Satay\t\t40.00\n");**

**printf("\t B\t\t30 Chicken Satay+20 Mutton Satay\t52.00\n");**

**printf("\t C\t\t10 Mutton Satay+40 Beef Satay\t\t46.00\n");**

**printf("\tAdd-On\t\t1 Ketupat\t\t\t\t0.60\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Input Combo Type\*/**

**printf("\n\n\t%d.Enter Combo Type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**/\*Loop\*/**

**while(comboType!='X')**

**{**

**/\*Input Quantity\*/**

**printf("\tEnter quantity\t\t\t\t:");**

**scanf("%d",&Qty);**

**fflush(stdin);**

**/\*Call Function\*/**

**price=calculatePrice(comboType);**

**/\*Calculate Order Price\*/**

**ordPrice=price\*Qty;**

**/\*Display Order Price\*/**

**printf("\tOrder Price for combo %c\t\t\t:RM%.2f\n\n",comboType,ordPrice);**

**/\*Accumulate Total Order Price\*/**

**totalPrice+=ordPrice;**

**/\*Increase counter by one\*/**

**counter++;**

**/\*Input Next Combo Type\*/**

**printf("\t%d.Enter combo type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**}**

**/\*Input Add-On\*/**

**printf("\n.\tAdd-on Ketupat(Y/N)\t\t\t:");**

**scanf("%c",&addOn);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**addOn=toupper(addOn);**

**/\*Input Add-On Quantity\*/**

**printf("\tEnter Ketupat quantity\t\t\t:");**

**scanf("%d",&addQty);**

**/\*Selection\*/**

**if(addOn=='Y')**

**addPrice=0.60;**

**else**

**addPrice=0.00;**

**/\*Calculate Order Price\*/**

**addOrd=addPrice\*addQty;**

**/\*Display Add-On\*/**

**printf("\tOrder Price for ketupat\t\t\t:Rm%.2f\n\n",addOrd);**

**/\*Calculate Total Order Price\*/**

**totalPrice+=addOrd;**

**/\*Display Total Order Price\*/**

**printf("\tTotal Order Price\t\t\t:RM%.2f",totalPrice);**

**getch ();**

**return 0;**

**}**

**/\*Function Name:calculatePrice\*/**

**/\*Parameter:char comboType\*/**

**/\*Return:float\*/**

**float calculatePrice(char comboType)**

**{**

**/\*Declaration\*/**

**float price;**

**/\*Selection\*/**

**if(comboType=='A')**

**price=40.00;**

**else if(comboType=='B')**

**price=52.00;**

**else if(comboType=='C')**

**price=46.00;**

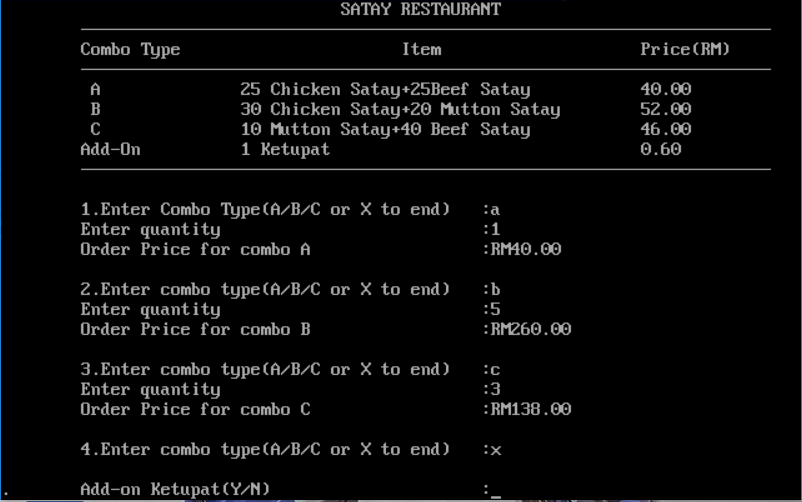
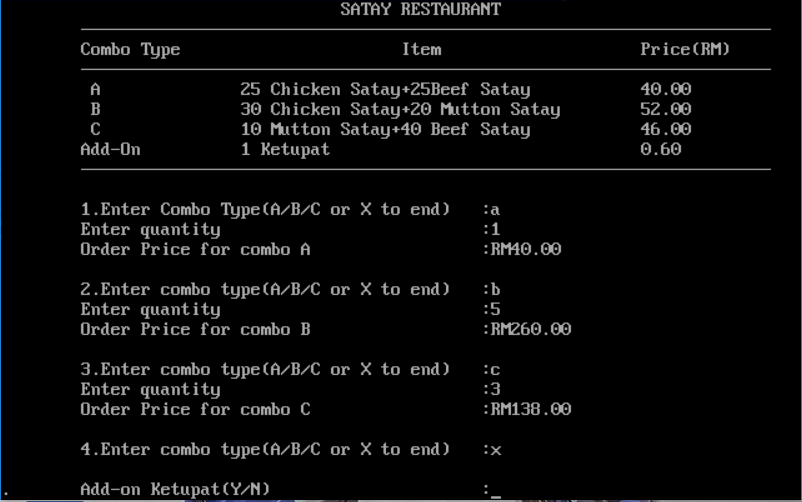
**else**

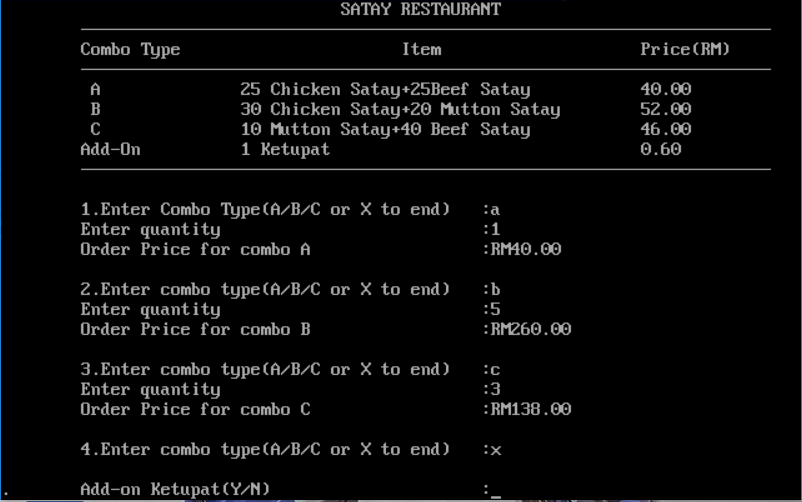
**price=0.00;**

**return price;**

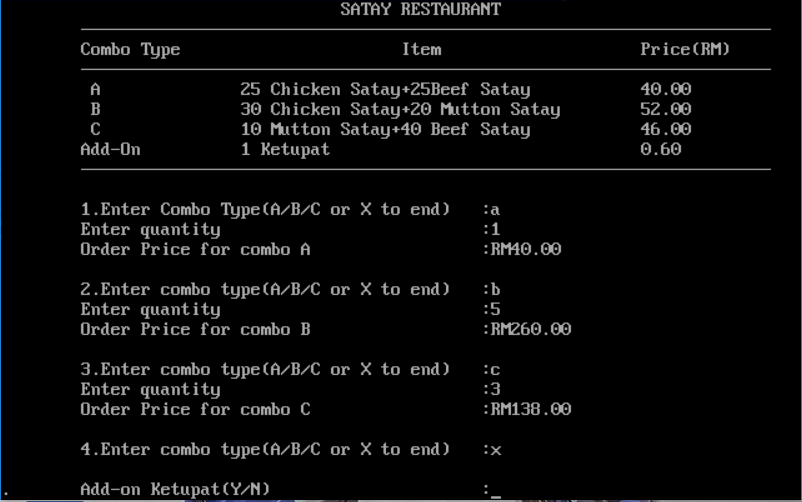
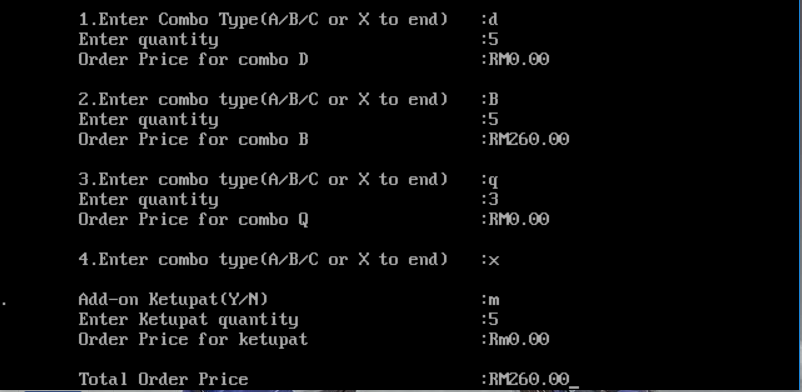
**}**

**Input and Output for Task B3**

* Sample Input and Output with ***correct*** combo type



* Sample Input and Output with ***wrong*** combo type



**Questions for Task B4**

Based on Assignment B3, construct an algorithm that calculates and stores the total order price for each customer in an array.

The algorithm will then display the total order price for each customer and the grand total due from all customers.

Use the following declaration in the program

float customerPayment[3];

**FlowChart for Task B4**

Start

False

A

F

Display customer

G

True

customer<N

?

customer=0

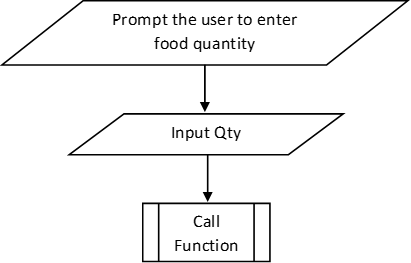
Display Selection List

grandTotal=0

Declare variable comboType, Qty, price, ordPrice, addOn, addQty, addPrice, counter, addOrd, customer, customerPayment[N] and grandTotal.

Declare function prototype calculatePrice

Define N to 3

****

False

B

C

D

comboType!=’X’

?

True

Prompt the user to enter first combo type

comboType=toupper(comboType)

Input comboType

customerPayment[customer]=0

counter=1

A

customerPayment[customer]=customerPayment[customer]+ordPrice

C

ordPrice=price\*Qty

Display ordPrice

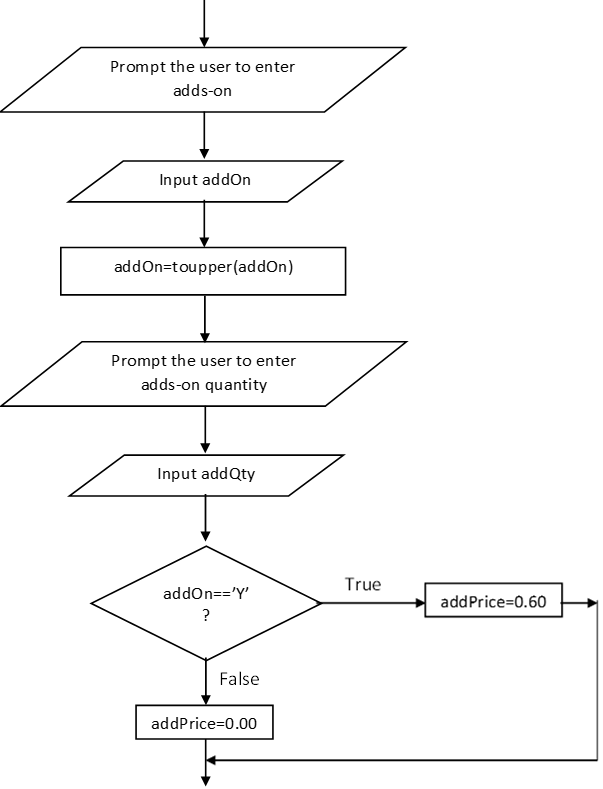
counter=counter+1

Prompt the user to enter next combo type

Input comboType

comboType=toupper(comboType)

B

****

D

addOrd=addQty\*addPrice

Display addOrd

E

customerPayment[customer]=customerPayment[customer]+addOrd

Display customerPayment[customer]

F

customer=customer+1

grandTotal=grandTotal+customerPayment[customer]

E

True

Display customer and customerPayment[customer]

customer=customer+1

End

Display grandTotal

G

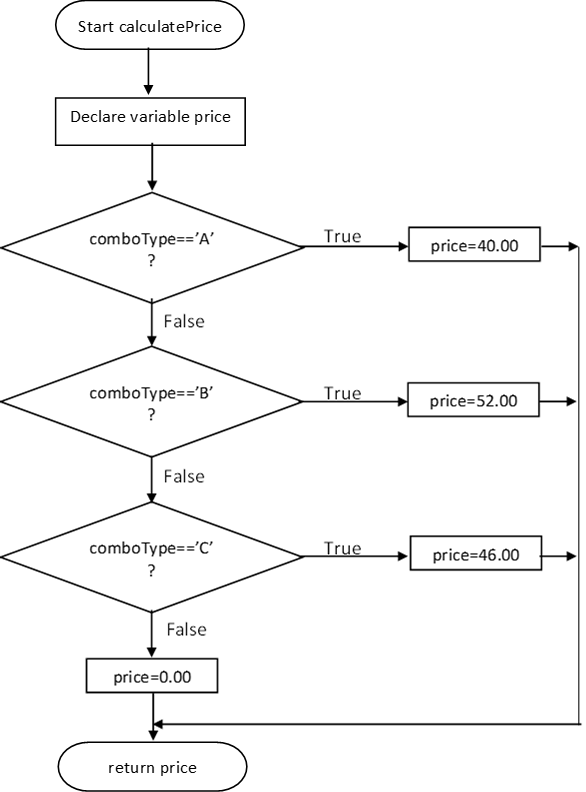
Display Order Price Summary

customer=0

customer<N

?

False

****

**Source Code for Task B4**

**/\*Name:Ng Ke Rong\*/**

**/\*I/C No:990124-07-5449\*/**

**/\*Date:2July2018\*/**

**#define N 3**

**#include <stdio.h>**

**int main()**

**{**

**/\*Function Prototype\*/**

**float calculatePrice(char comboType);**

**/\*Declaration\*/**

**float customerPayment[N];**

**char comboType;**

**int Qty;**

**char addOn;**

**int addQty;**

**float ordPrice;**

**float price;**

**float addPrice;**

**int counter;**

**int Line;**

**float addOrd;**

**float grandTotal;**

**int customer;**

**/\*Initialize\*/**

**grandTotal=0;**

**/\*Display Selection List\*/**

**clrscr();**

**printf("\t\t\t\tSATAY RESTAURANT\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tCombo Type\t\t\tItem\t\t\tPrice(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\t A\t\t25 Chicken Satay+25Beef Satay\t\t40.00\n");**

**printf("\t B\t\t30 Chicken Satay+20 Mutton Satay\t52.00\n");**

**printf("\t C\t\t10 Mutton Satay+40 Beef Satay\t\t46.00\n");**

**printf("\tAdd-On\t\t1 Ketupat\t\t\t\t0.60\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Loop\*/**

**for(customer=0;customer<N;customer++)**

**{**

**/\*Input Customer\*/**

**printf("\n\tCustomer%2d:\n\t",customer+1);**

**Line=1;**

**while(Line<12)**

**{**

**printf("-");**

**Line++;**

**}**

**/\*Initialize\*/**

**counter=1;**

**customerPayment[customer]=0;**

**/\*Input First Combo Type\*/**

**printf("\n\t\t%d.Enter Combo Type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**/\*Loop\*/**

**while(comboType!='X')**

**{**

**/\*Input Quantity\*/**

**printf("\t\t Enter quantity\t\t\t:");**

**scanf("%d",&Qty);**

**fflush(stdin);**

**/\*Call Function\*/**

**price=calculatePrice(comboType);**

**/\*Calculate Order Price\*/**

**ordPrice=price\*Qty;**

**/\*Display Order Price\*/**

**printf("\t\t Order Price for combo %c\t\t:RM%.2f\n\n",comboType,ordPrice);**

**/\*Accumulate Total Order Price\*/**

**customerPayment[customer]+=ordPrice;**

**/\*Increase counter by one\*/**

**counter++;**

**/\*Input Next Combo Type\*/**

**printf("\t\t%d.Enter combo type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**}**

**/\*Input Add-On\*/**

**printf("\n\tAdd-on Ketupat(Y/N)\t\t\t\t:");**

**scanf("%c",&addOn);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**addOn=toupper(addOn);**

**/\*Input Adds-On Quantity\*/**

**printf("\tEnter Ketupat quantity\t\t\t\t:");**

**scanf("%d",&addQty);**

**fflush(stdin);**

**/\*Selection\*/**

**if(addOn=='Y')**

**addPrice=0.60;**

**else**

**addPrice=0.00;**

**/\*Calculate Order Price\*/**

**addOrd=addPrice\*addQty;**

**/\*Display Add-On\*/**

**printf("\tOrder Price for ketupat\t\t\t\t:Rm%.2f\n\n",addOrd);**

**/\*Calculate Total Order Price\*/**

**customerPayment[customer]+=addOrd;**

**/\*Display Total Price\*/**

**printf("Total order price for customer%2d\t\t:Rm%.2f\n\t",customer+1,customerPayment[customer]);**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Accumulate Grand Total\*/**

**grandTotal+=customerPayment[customer];**

**}**

**/\*Display Order Price Summary\*/**

**printf("\n\n\t\t\tOrder Price Summary\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tCustomer No\t\t\tTotal Order Price(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Loop\*/**

**for(customer=0;customer<N;customer++)**

**{**

**/\*Display Customer and Total Order Price\*/**

**printf("\n\t\t%d\t\t\t\t%.2f",customer+1,customerPayment[customer]);**

**}**

**printf("\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Display Grand Total\*/**

**printf("\n\tGrand Total\t\t\t\t%.2f\n\t",grandTotal);**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",209);**

**Line++;**

**}**

**getch ();**

**return 0;**

**}**

**/\*Function Name:calculatePrice\*/**

**/\*Parameter:char comboType\*/**

**/\*Return:float\*/**

**float calculatePrice(char comboType)**

**{**

**/\*Declaration\*/**

**float price;**

**/\*Selection\*/**

**if(comboType=='A')**

**price=40.00;**

**else if(comboType=='B')**

**price=52.00;**

**else if(comboType=='C')**

**price=46.00;**

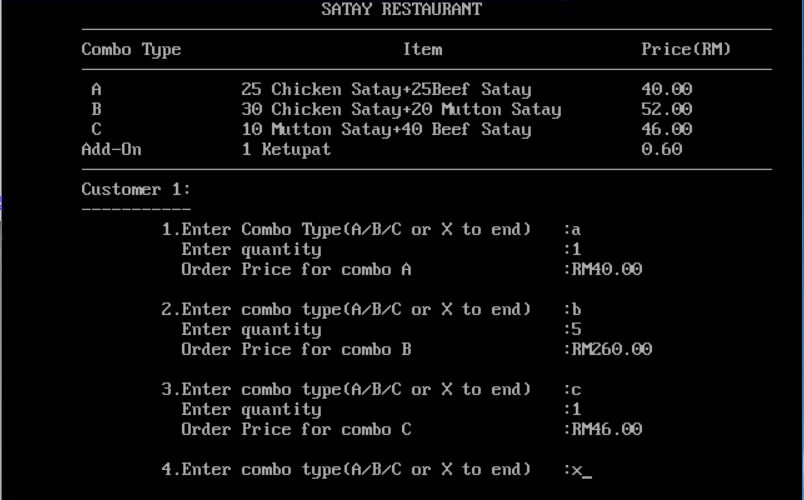
**else**

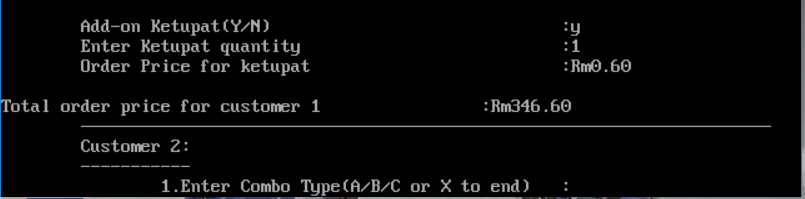
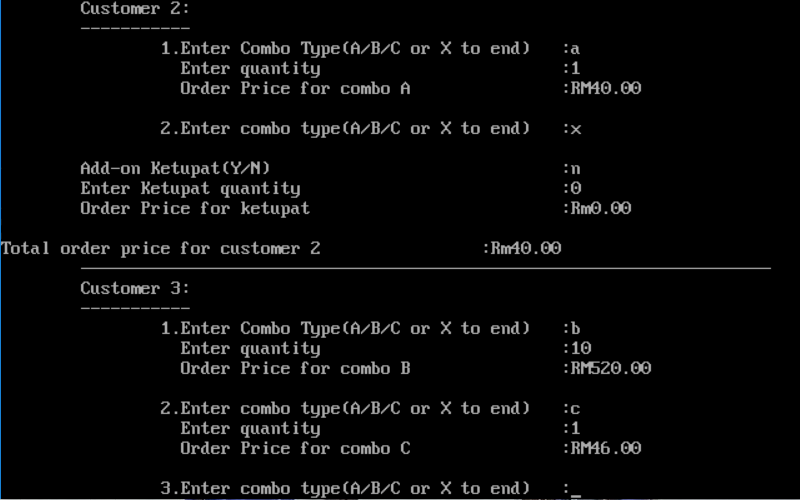
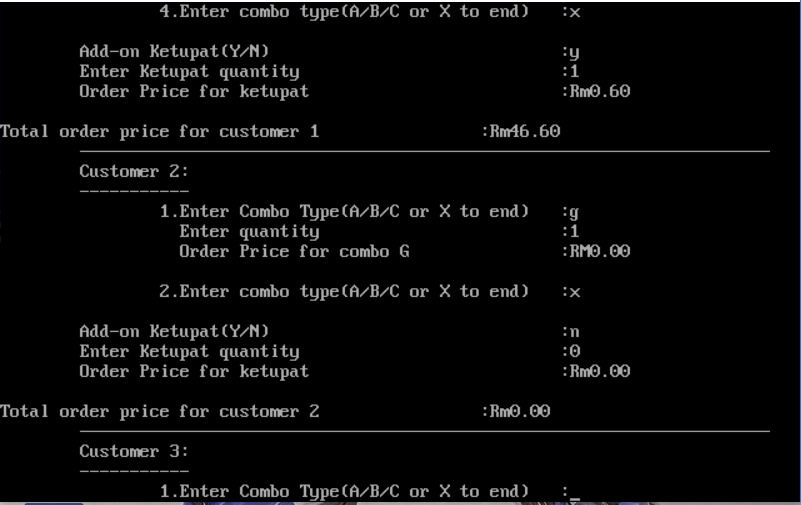
**price=0.00;**

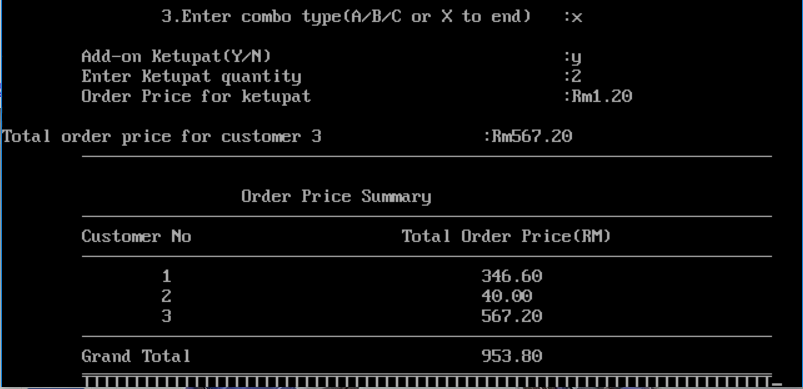
**return price;**

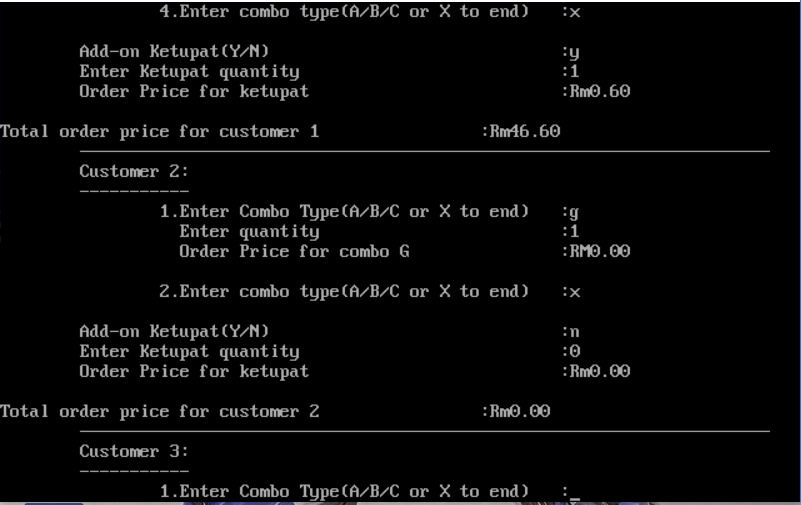
**}**

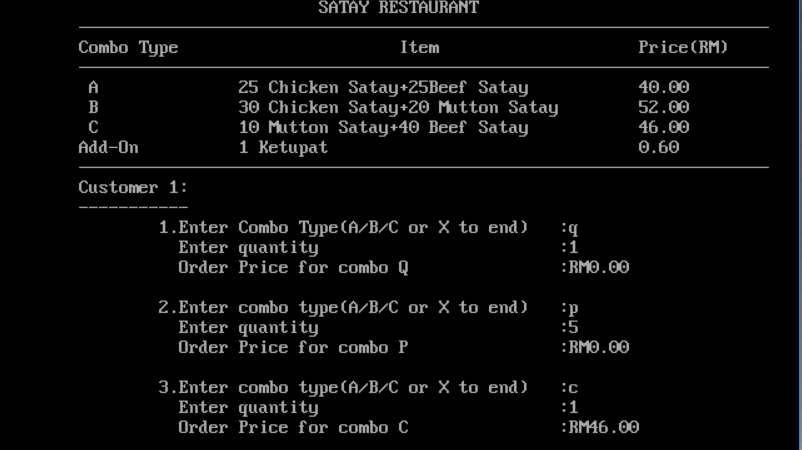
**Input and Output for Task B4**

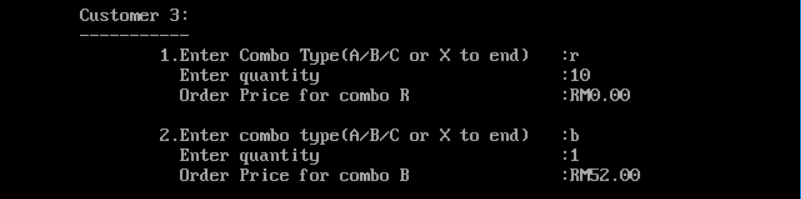
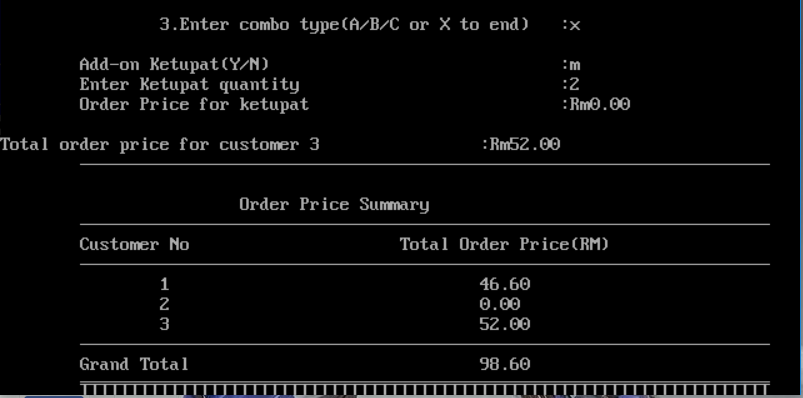
* Sample Input and Output with ***correct*** combo type





* Sample Input and Output with ***wrong*** combo type





**Questions for Task B5**

Based on Assignment B4, construct an algorithm to read and store records for three customers using the following structure

struct customer

{

int customerID;

char customerName[SIZE];

char customerPhoneNo[12];

float customerOrderPrice;

};

typedef struct customer customerInfo;

The algorithm will then display customer identification number, name, phone number, total order price for each customer and the grand total due from all customers.

Use the following declaration in the program

float customerPayment[3];

**FlowChart for Task B5**

Start

A

grandTotal=0

Display Selection List

Declare variable comboType, Qty, price, ordPrice, addOn, addQty, addPrice, counter, addOrd, customer, customerPayment[N], grandTotal and customerInfo customerList[N].

Declare function prototype calculatePrice

Declare struct customer

{

int customerID;

char customerName[SIZE];

char customerPhoneNo[12];

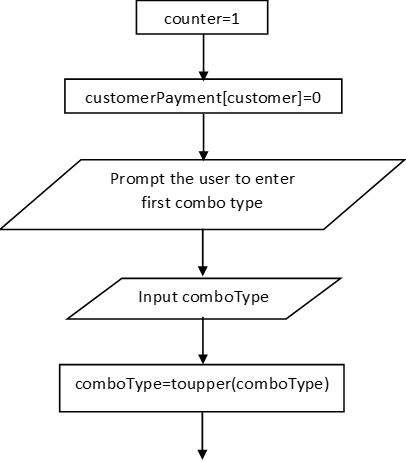
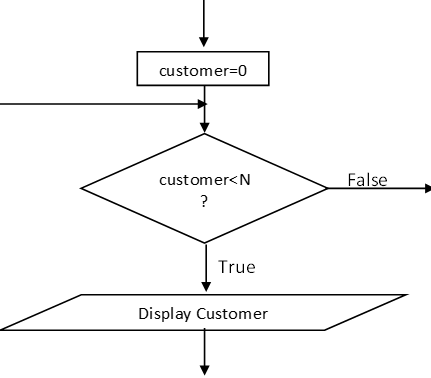
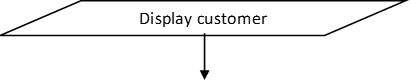
float customerOrderPrice;

}

typedef struct customer customerInfo;

Define SIZE to 12

Define N to 3

****

B

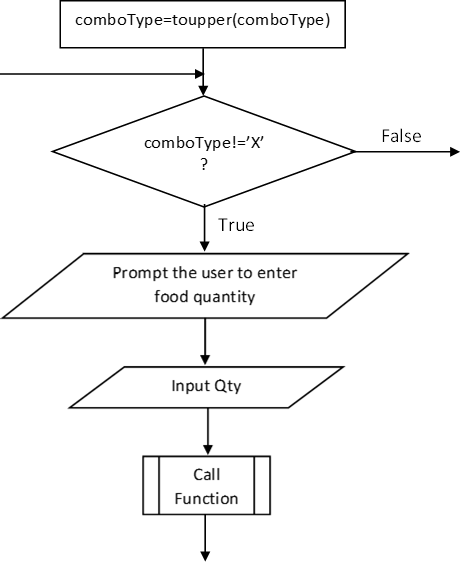
H

G

Input customerList[customer]

Prompt the user to enter customer list

A

****

C

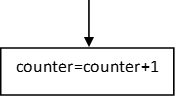
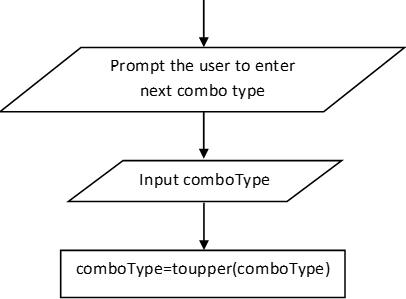
ordPrice=price\*Qty

Display ordPrice

E

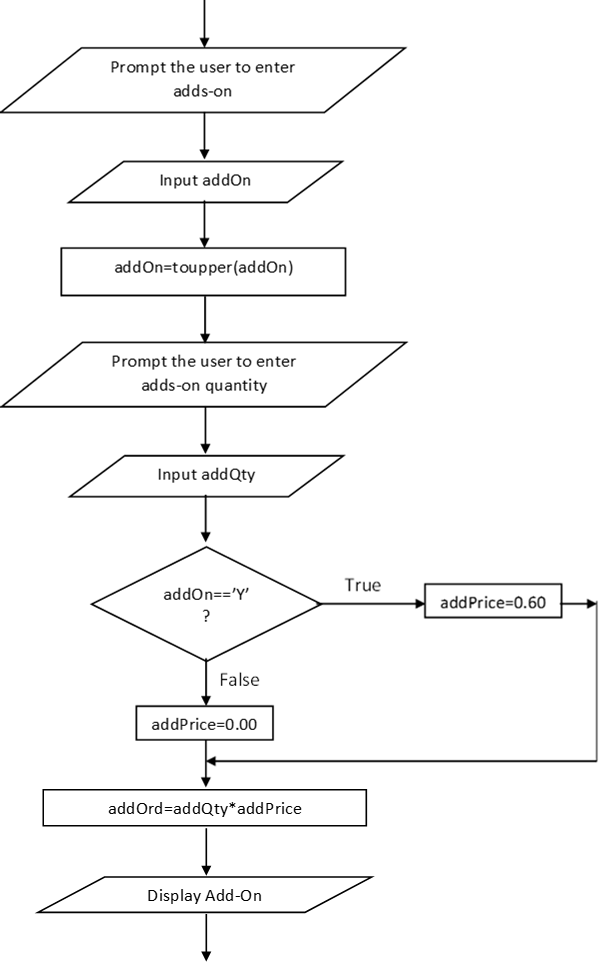
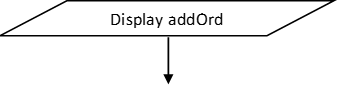
D

B

****

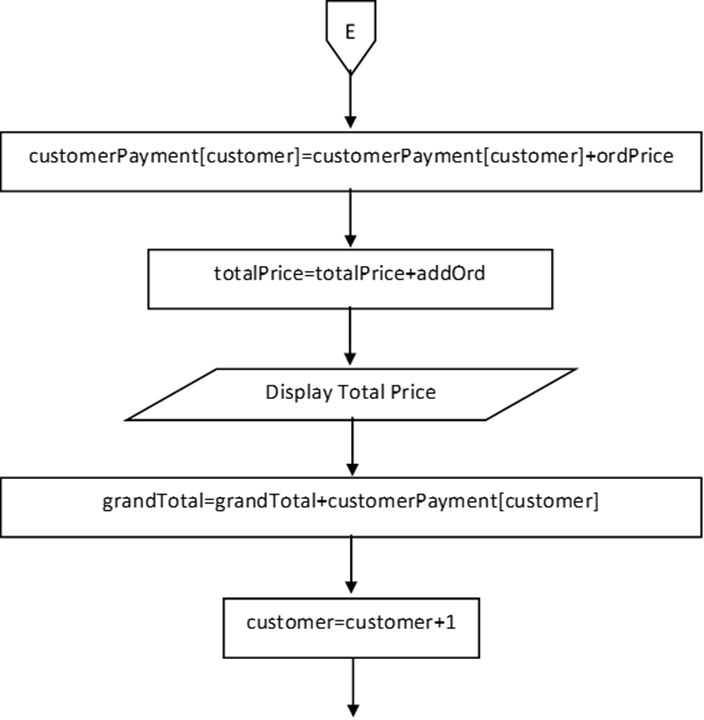
D

C

****

F

E

****

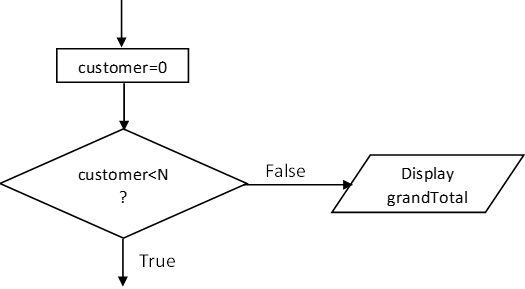
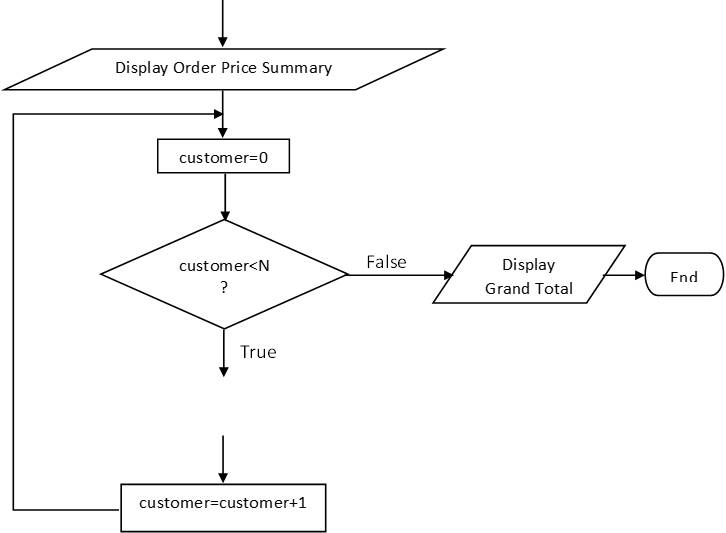
G

customerList[customer].customerOrderPrice=customerPayment[customer]

Display customerPayment[customer]

customerPayment[customer]=customerPayment[customer]+addOrd

F

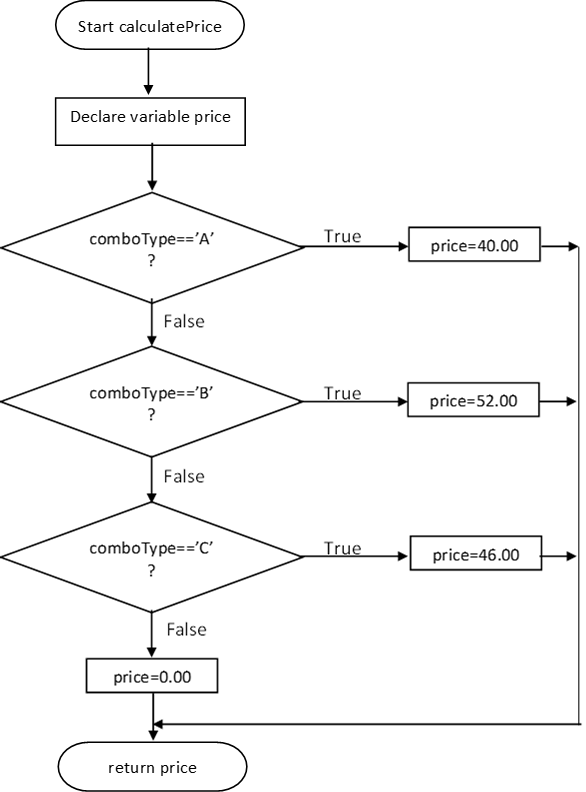
****

H

customer=customer+1

Display customerList[customer].customerID, customerList[customer].customerName, customerList[customer].customerPhoneNo and customerList[customer].customerOrderPrice.

End

****

**Source Code for Task B5**

**/\*Name:Ng Ke Rong\*/**

**/\*I/C No:990124-07-5449\*/**

**/\*Date:2July2018\*/**

**#include <stdio.h>**

**#define N 3**

**#define SIZE 20**

**struct customer**

**{**

**int customerID;**

**char customerName[SIZE];**

**char customerPhoneNo[12];**

**float customerOrderPrice;**

**};**

**typedef struct customer customerInfo;**

**int main()**

**{**

**/\*Function Prototype\*/**

**float calculatePrice(char comboType);**

**/\*Declaration\*/**

**float customerPayment[N];**

**customerInfo customerList[N];**

**char comboType;**

**int Qty;**

**char addOn;**

**int addQty;**

**float ordPrice;**

**float price;**

**float addPrice;**

**int counter;**

**int Line;**

**float addOrd;**

**float grandTotal;**

**int customer;**

**/\*Initialize\*/**

**grandTotal=0;**

**/\*Display Selection List\*/**

**clrscr();**

**printf("\t\t\t\tSATAY RESTAURANT\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tCombo Type\t\t\tItem\t\t\tPrice(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\t A\t\t25 Chicken Satay+25Beef Satay\t\t40.00\n");**

**printf("\t B\t\t30 Chicken Satay+20 Mutton Satay\t52.00\n");**

**printf("\t C\t\t10 Mutton Satay+40 Beef Satay\t\t46.00\n");**

**printf("\tAdd-On\t\t1 Ketupat\t\t\t\t0.60\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Loop\*/**

**for(customer=0;customer<N;customer++)**

**{**

**/\*Display Customer\*/**

**printf("\n\tCustomer%2d:\n\t",customer+1);**

**Line=1;**

**while(Line<12)**

**{**

**printf("-");**

**Line++;**

**}**

**/\*Initialize\*/**

**counter=1;**

**customerPayment[customer]=0;**

**/\*Input Customer Info\*/**

**printf("\n\n\t\tEnter Customer ID\t:");**

**scanf("%d",&customerList[customer].customerID);**

**printf("\t\tEnter Customer Name\t:");**

**scanf("%s",&customerList[customer].customerName);**

**printf("\t\tEnter Customer Phone No\t:");**

**scanf("%s",&customerList[customer].customerPhoneNo);**

**/\*Input First Combo Type\*/**

**printf("\n\n\t\t%d.Enter Combo Type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**/\*Loop\*/**

**while(comboType!='X')**

**{**

**/\*Input Quantity\*/**

**printf("\t\t Enter quantity\t\t\t:");**

**scanf("%d",&Qty);**

**fflush(stdin);**

**/\*Call Function\*/**

**price=calculatePrice(comboType);**

**/\*Calculate Order Price\*/**

**ordPrice=price\*Qty;**

**/\*Display Order Price\*/**

**printf("\t\t Order Price for combo %c\t\t:RM%.2f\n\n",comboType,ordPrice);**

**/\*Accumulate Total Order Price\*/**

**customerPayment[customer]+=ordPrice;**

**/\*Increase counter by one\*/**

**counter++;**

**/\*Input Next Combo Type\*/**

**printf("\t\t%d.Enter combo type(A/B/C or X to end)\t:",counter);**

**scanf("%c",&comboType);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**comboType=toupper(comboType);**

**}**

**/\*Input Add-On\*/**

**printf("\n\tAdd-on Ketupat(Y/N)\t\t\t\t:");**

**scanf("%c",&addOn);**

**fflush(stdin);**

**/\*Convert to Uppercase\*/**

**addOn=toupper(addOn);**

**/\*Input Adds-On Quantity\*/**

**printf("\tEnter Ketupat quantity\t\t\t\t:");**

**scanf("%d",&addQty);**

**fflush(stdin);**

**/\*Selection\*/**

**if(addOn=='Y')**

**addPrice=0.60;**

**else**

**addPrice=0.00;**

**/\*Calculate Order Price\*/**

**addOrd=addPrice\*addQty;**

**/\*Display Add-On\*/**

**printf("\tOrder Price for ketupat\t\t\t\t:Rm%.2f\n\n",addOrd);**

**/\*Calculate Total Order Price\*/**

**customerPayment[customer]+=addOrd;**

**/\*Display Total Price\*/**

**printf("Total order price for customer%2d\t\t\t:RM%.2f\n\t",customer+1,customerPayment[customer]);**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Assign Customer Order Price\*/**

**customerList[customer].customerOrderPrice=customerPayment[customer];**

**/\*Accumulate Grand Total\*/**

**grandTotal+=customerPayment[customer];**

**}**

**/\*Display Order Price Summary\*/**

**printf("\n\n\t\t\t\tOrder Price Summary\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**printf("\n\tID\tCustomer Name\tCustomer Phone No\tTotal Order Price(RM)\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Loop\*/**

**for(customer=0;customer<N;customer++)**

**{**

**/\*Display Customer Info\*/**

**printf("\n\t%d\t",customerList[customer].customerID);**

**printf("%s\t\t",customerList[customer].customerName);**

**printf("%s\t\t",customerList[customer].customerPhoneNo);**

**printf("%.2f",customerList[customer].customerOrderPrice);**

**}**

**printf("\n\t");**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**/\*Display Grand Total\*/**

**printf("\n\t\t\t\t\tGrand Total:\t%.2f\n\t",grandTotal);**

**Line=1;**

**while(Line<70)**

**{**

**printf("%c",196);**

**Line++;**

**}**

**getch ();**

**return 0;**

**}**

**/\*Function Name:calculatePrice\*/**

**/\*Parameter:char comboType\*/**

**/\*Return:float\*/**

**float calculatePrice(char comboType)**

**{**

**/\*Declaration\*/**

**float price;**

**/\*Selection\*/**

**if(comboType=='A')**

**price=40.00;**

**else if(comboType=='B')**

**price=52.00;**

**else if(comboType=='C')**

**price=46.00;**

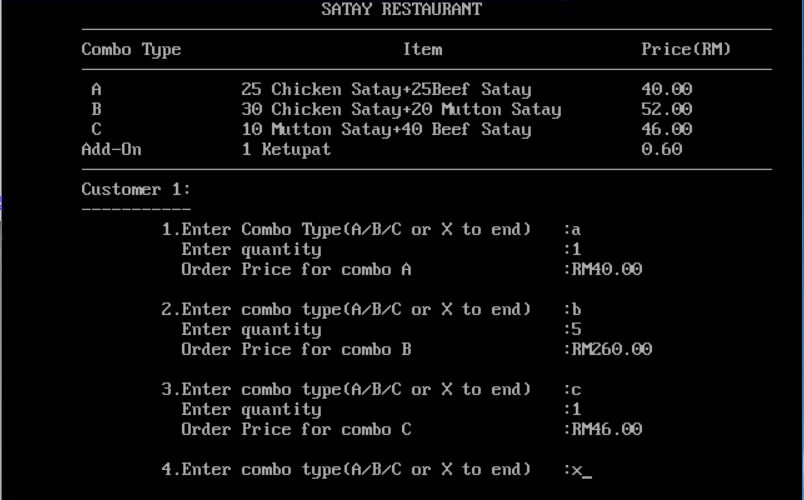
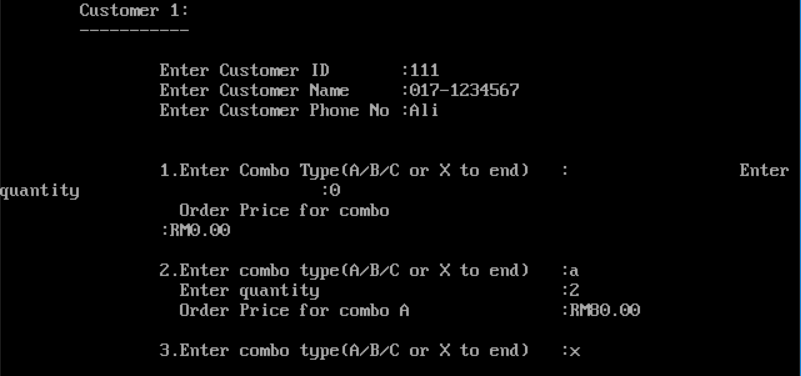
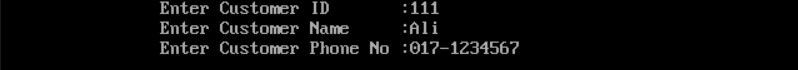
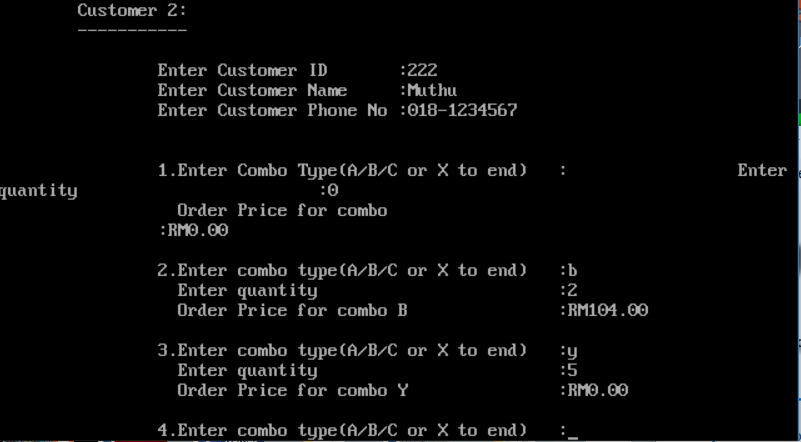
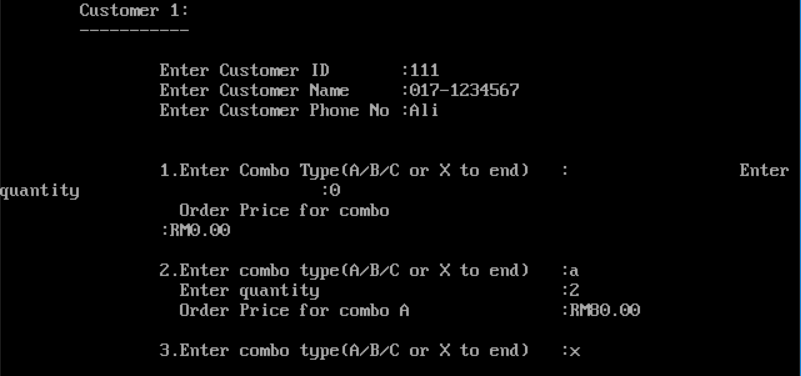
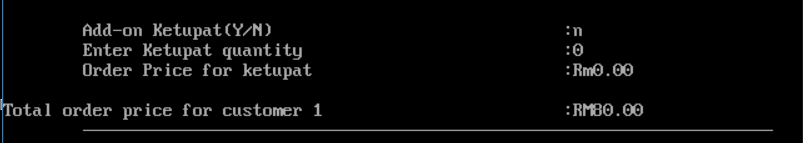
**else**

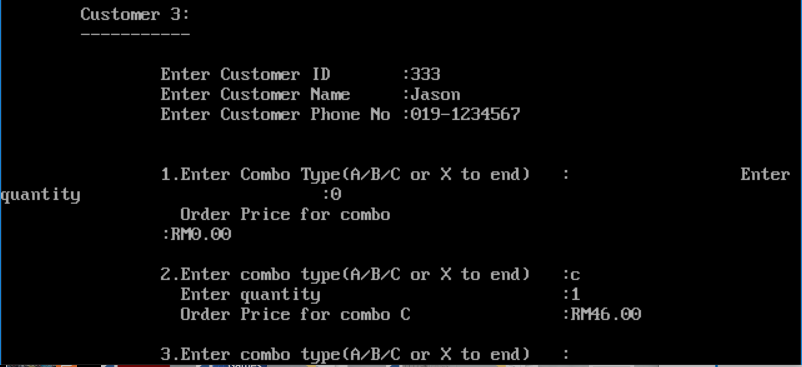
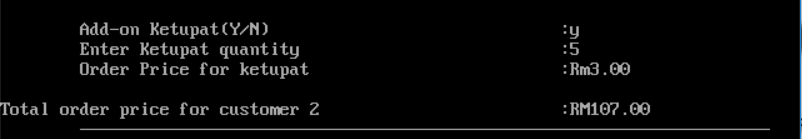
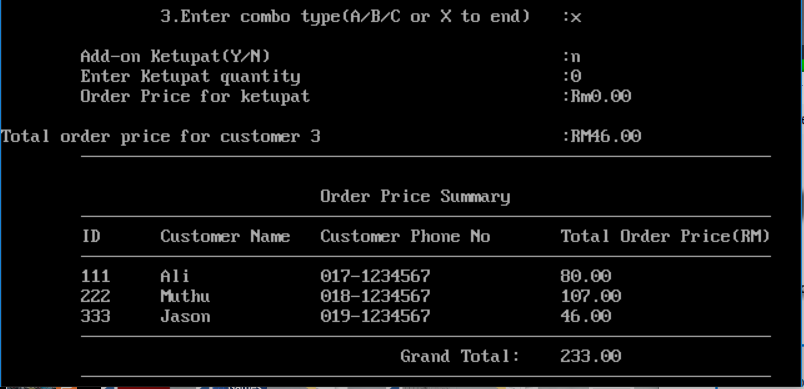
**price=0.00;**

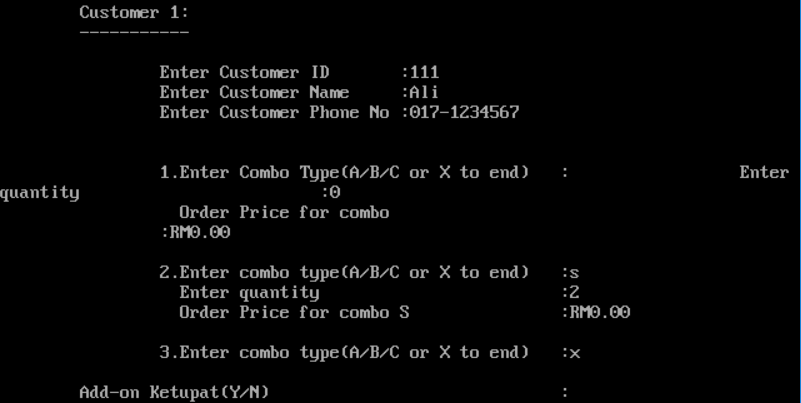
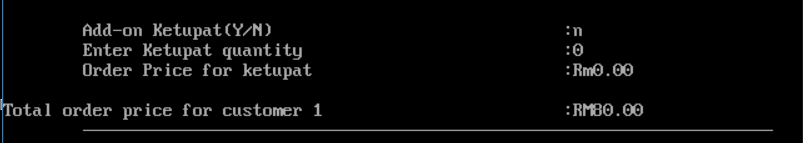
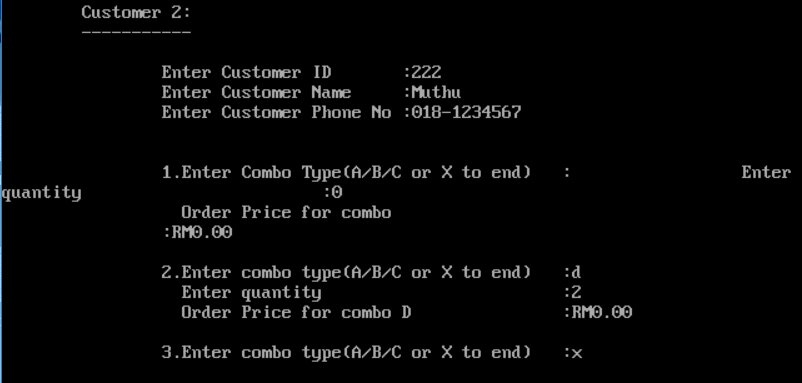
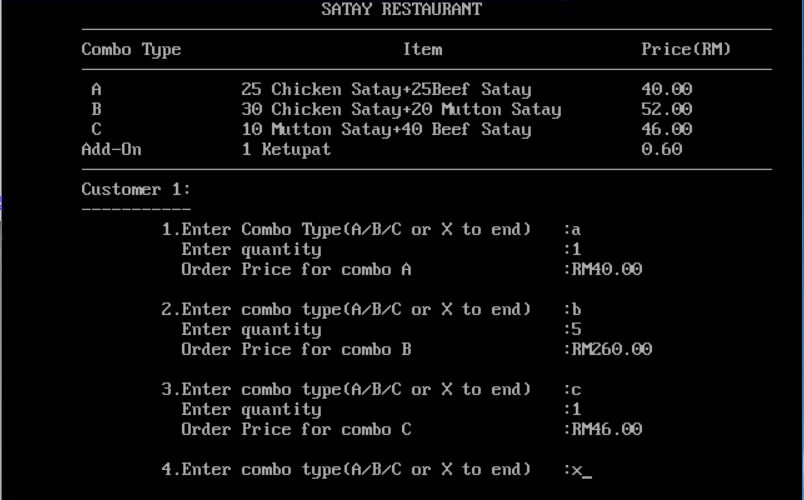
**return price;**

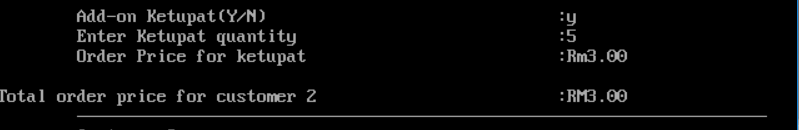
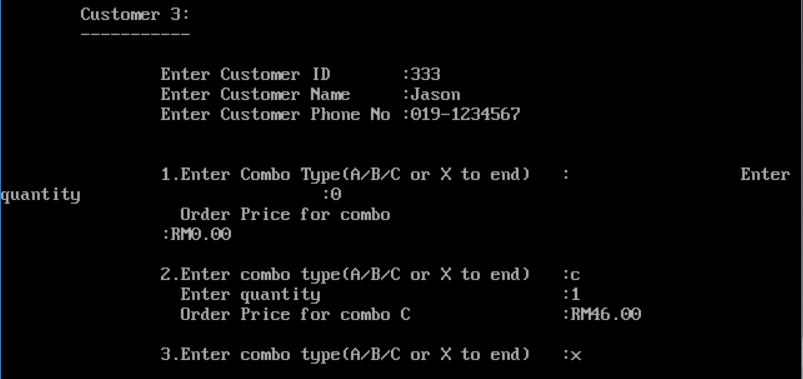
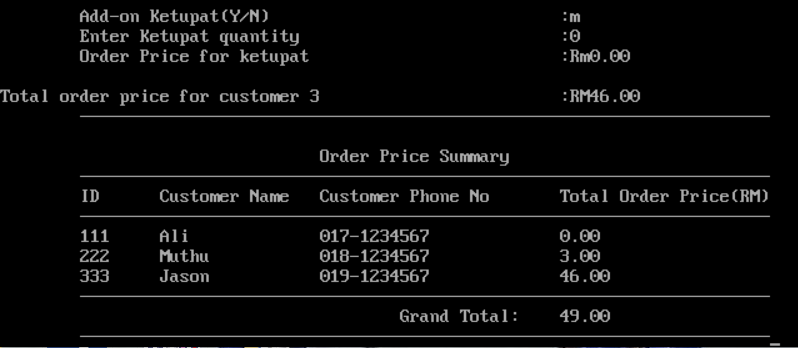
**}**

**Input and Output for Task B5**

* Sample Input and Output with ***correct*** combo type



* Sample Input and Output with ***wrong*** combo type



**Closing Mark**

After complete this Assignment B, I feel that C program is easy as ABC to learn. In this assignment, I learn how to create a menu, how a program run a program, how to write a program and debug errors in a program. Program can help me in future to create an excellent apps.