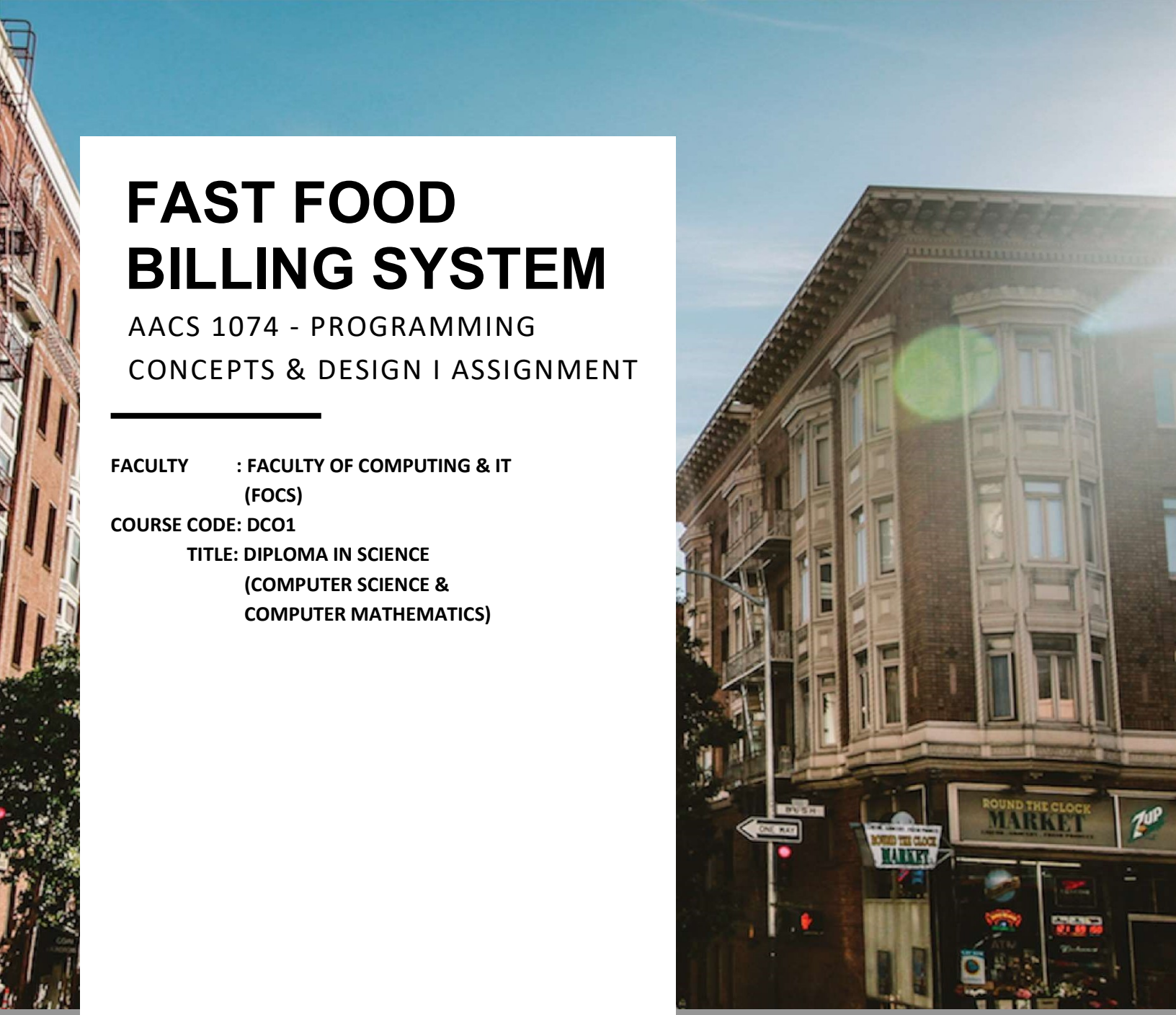


FAST FOOD BILLING SYSTEM

AACS 1074 - PROGRAMMING CONCEPTS & DESIGN I ASSIGNMENT

FACULTY : FACULTY OF COMPUTING & IT (FOCS)
COURSE CODE: DCO1
TITLE: DIPLOMA IN SCIENCE (COMPUTER SCIENCE & COMPUTER MATHEMATICS)

AUGUST 9

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
Authored by: Hiew Long Shun (DCO1-G5)
Tutor : Mr. Tan Cheng Tien

TARC
TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE

Declaration of Originality

Signature: _____ Name: _____ Date: _____

Table of Content

Description & Purpose

This program is primarily aimed at fast food restaurant cashiers. Besides delivering fast food, a fast food restaurant needs to bill their customers efficiently to ensure that customer satisfaction is kept at maximum level and to maximize revenue. It is also of vital importance to keep records of the different types of foods used up. This is to keep track of the distribution of sales, allowing a fast food restaurant chain to perform efficient restocking to prevent waste, and to make informed decisions on the food to be offered to customers in the future.

This program is made to allow restaurants to bill customers faster and more efficiently, increasing customer flow and profits, and at the end of the day produce a sales report to aid in business decisions.

A research done by QSR Magazine shown that the average time for a drive-thru order among fast food chains is 224.77 seconds, with Raising Cane's leading at 168.23 seconds, managing about 7 more customers per hour. Therefore, efficient billing can cut down the time taken by customers in queue, which is a crucial part to ensuring a fast food restaurant's success.

Added Features

1. **Order Cancellation Function**
    a. To cancel an order, the cashier needs to keep track of the combo and quantity combination ordered by the customer who cancelled an order, then key in the negative matching quantity for each combo to reverse the order made.
    b. This poses two challenges, first is the speed at which cancellation can be done. If a customer orders large quantity of orders in no order, making changes will take some time. During this period, the cashier counter cannot take another other without affecting the accuracy of daily sales report. This severely cripples the ability of a restaurant to handle large influx of customers, losing sales and revenue. Second is human mistakes, a cashier can easily reverse wrongly due to an error, which affects the accuracy of the daily sales report.
    c. This function allows a cashier to instantly cancel off an order with just two keypresses and supports cancelling in both the taking orders and billing phase to instantly cancels order, preventing system downtime**.**

2. **Write Daily Sales Report to Text File**
    a. At the end of the day, a daily sales report is generated on-screen. However, this report generally needs to be saved to keep track of sales, perform accounting, and to make future financial decisions.
    b. Without a proper standard for recording the daily sales report to a file, making use of the daily sales report to improve a fast food restaurant will be difficult and messy.
    c. This function automates generation of a text file with a file name of "Daily-Sales-Report-DD-MM-YYYY.txt" and writes the daily sales report in a consistent format to the text file. This allows user of the sales report to make use of it easily and effectively.

3. **Make sure amount paid by customer is enough for order**
    a. Sometimes, a customer might accidentally pay the wrong amount that is lower than the net total for his order, which is the total after taxes, also known as amount payable. In this case, it will slow down the cashier because the cashier will need to count them and figure out whether if the amount is enough.
    b. Moreover, it is possible for the cashier to miss out some details when attempting to retrieve the remaining amount from the customer.
    c. This feature will make sure that the cashier is able to quickly let customer know whether the amount paid is enough or not, by only asking for the amount paid, and performing the decision whether the paid amount is enough or not. If it is not enough, it will prompt again for an indefinite number of times until the sufficient amount of money is received.
    d. This way, the cashier will have to do less calculation, and put more focus into delivering a faster and more efficient service.

Program Testing & Outputs

Run 1

Run 1 scenario: More than 1 customer, same combo ordered twice by same customer,
1 customer cancelled order **halfway through ordering,**
report not generated to a file
(Sales Report should automatically be displayed after last customer)

Inputs						**Expected Results / Outputs**			
Cust #	Combo ordered	Qty	Payment	Cancel?	Write to file?	Combo charges	SST 10%	Total charges	Change to be given
1	D C D	2 1 3	160			24 * 2 = 48 18 * 1 = 18 24 * 3 = 72 — 138	13.80	151.80	8.20
2	D A C	4 2 1	0	Yes			0	0	0
					No				
						Total Customers: 1 Total Combo Sales: A 0 * 10 = 0 B 0 * 15 = 0 C 1 * 18 = 18 D 5 * 24 = 120 ---- ---- 6 138	13.80	151.80	
						No Daily-Sales-Report-DD-MM-YYYY.txt created in present working directory			

```
 ____  ____  ___   ____  __  ____  ____  ____  __   __  ____
/ ___)(  __)/ __) (  _ \/ _\ / ___)(_  _)(  _ \/  \ /  \(    \
\___ \ ) _)( (__   ) _ /    \\___ \  )(   ) _ (  O )  O )) D (
(____/(____)\___) (__) \_/\_/(____/ (__) (____/\__/ \__/(____/

You are customer: 1

----MENU----
Combo Contents                                                                                  Price (RM)

A     Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola Drink (L)                  10.00
B     Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink (L)                        15.00
C     2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)                                         18.00
D     3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2 Pepsi drink (S)             24.00

----ORDERING----
Combo A, B, C, D (other = exit, X = cancel): D
Quantity        : 2
        Combo D x 2 @ RM    24.00 = RM     48.00
Combo A, B, C, D (other = exit, X = cancel): C
Quantity        : 1
        Combo C x 1 @ RM    18.00 = RM     18.00
Combo A, B, C, D (other = exit, X = cancel): D
Quantity        : 3
        Combo D x 3 @ RM    24.00 = RM     72.00
Combo A, B, C, D (other = exit, X = cancel): o

----BILLING----
        Gross total                    = RM   138.00
        SST total                      = RM    13.80
        TOTAL PAYABLE                  = RM   151.80
        Amount Paid (<-999 = cancel)   = RM    160
        Change                         = RM     8.20

Thank you, have a nice day and see you later!

Next customer? (y/n): y

You are customer: 2

----MENU----
Combo Contents                                                                                  Price (RM)

A     Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola Drink (L)                  10.00
B     Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink (L)                        15.00
C     2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)                                         18.00
D     3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2 Pepsi drink (S)             24.00

----ORDERING----
Combo A, B, C, D (other = exit, X = cancel): D
Quantity        : 4
        Combo D x 4 @ RM    24.00 = RM     96.00
Combo A, B, C, D (other = exit, X = cancel): A
Quantity        : 2
        Combo A x 2 @ RM    10.00 = RM     20.00
Combo A, B, C, D (other = exit, X = cancel): C
Quantity        : 1
        Combo C x 1 @ RM    18.00 = RM     18.00
Combo A, B, C, D (other = exit, X = cancel): X

----Cancellation----

Order cancelled, customer count - 1.


Thank you, have a nice day and see you later!

Next customer? (y/n): n
```

```
----DAILY SALES REPORT----
Total customers for today: 1

Combo Sales For Today

Combo      Quantity Sold    Sales Amount
  A                    0            0.00
  B                    0            0.00
  C                    0            0.00
  D                    0            0.00

======            ======         =======
TOTALS                 0          138.00

        TOTAL SST charges    =     13.80
        TOTAL RM collected   =    151.80

Do you want to write the daily sales report to a file? (y/n): n

----ENDING----
~Thank you for using SFC Fast Food's Fast Food Billing System (F2BS)~
Made by: Hiew LS
--------------
Press any key to continue . . .
```

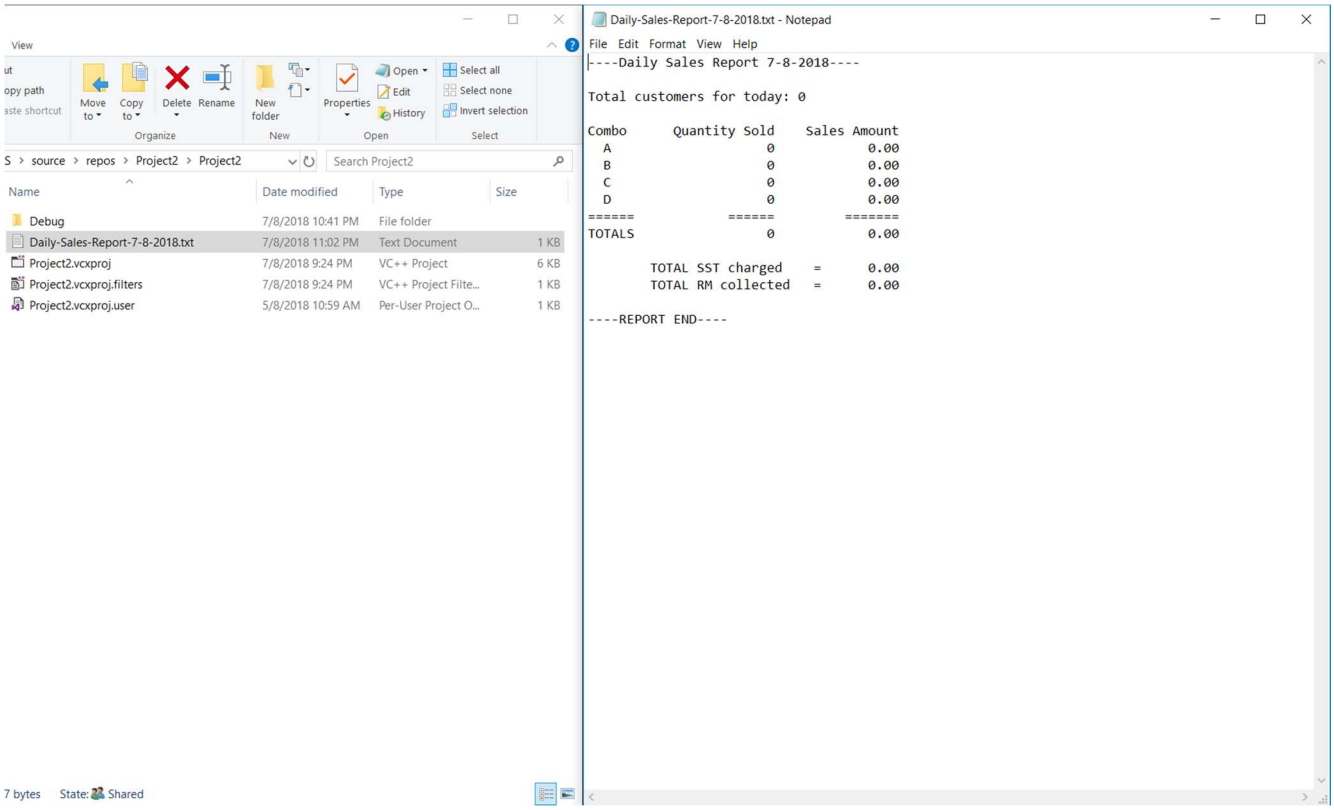
Run 2

Run 2 scenario: 1 customer, 1 customer ordered 0 sets of combo D,
1 customer cancelled order **halfway through billing**,
report generated to a file
(Sales Report should automatically be displayed after last customer)

Inputs						**Expected Results / Outputs**			
Cust #	Combo ordered	Qty	Payment	Cancel?	Write to file?	Combo charges	SST 10%	Total charges	Change to be given
1	A	4	250	Yes		10 * 4 = 40 15 * 8 = 120 18 * 2 = 36 18 * 0 = 0 — 196	19.60	215.60	
	B	8							
	C	2							
	D	0							
					Yes				
						Total Customers: 1 Total Combo Sales: A 4 * 10 = 40 B 8 * 15 = 120 C 2 * 18 = 36 D 0 * 24 = 0 ---- ---- 14 196	19.60	215.60	

Daily-Sales-Report-DD-MM-YYYY.txt created in present working directory

```
 ____  ____  ___   ____  __  ____  ____  ____  __    __  ____
/ ___)(  _ \/ __) (  _ \/ _\(  _ \(  _ \(  _ \(  )  (  )(    \
\___ \ ) _ ( (__   ) _ /    \) __/ ) _ ( )   / )(__  )(  ) D (
(____/(____/\___) (____/\_/\_(__)  (____/(__\_)(____)(__)(____/

You are customer: 1

----MENU----
Combo Contents                                                                                  Price (RM)

A     Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola Drink (L)                  10.00
B     Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink (L)                        15.00
C     2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)                                         18.00
D     3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2 Pepsi drink (S)             24.00

----ORDERING----
Combo A, B, C, D (other = exit, X = cancel): A
Quantity        : 4
        Combo A x 4 @ RM    10.00 = RM     40.00
Combo A, B, C, D (other = exit, X = cancel): B
Quantity        : 8
        Combo B x 8 @ RM    15.00 = RM    120.00
Combo A, B, C, D (other = exit, X = cancel): C
Quantity        : 2
        Combo C x 2 @ RM    18.00 = RM     36.00
Combo A, B, C, D (other = exit, X = cancel): D
Quantity        : 0
        Combo D x 0 @ RM    24.00 = RM      0.00
Combo A, B, C, D (other = exit, X = cancel): o

----BILLING----
        Gross total                     = RM   196.00
        SST total                       = RM    19.60
        TOTAL PAYABLE                   = RM   215.60
        Amount Paid (<-999 = cancel)    = RM    -999

----Cancellation----

Order cancelled, customer count - 1.


Thank you, have a nice day and see you later!

Next customer? (y/n): n


----DAILY SALES REPORT----
Total customers for today: 0

Combo Sales For Today

Combo      Quantity Sold    Sales Amount
  A                    0            0.00
  B                    0            0.00
  C                    0            0.00
  D                    0            0.00

======            ======         =======
TOTALS                 0            0.00

        TOTAL SST charges    =      0.00
        TOTAL RM collected   =      0.00

Do you want to write the daily sales report to a file? (y/n): y
Successfully wrote to: .\Daily-Sales-Report-7-8-2018.txt

----ENDING----
~Thank you for using SFC Fast Food's Fast Food Billing System (F2BS)~
Made by: Hiew LS
--------------
Press any key to continue . . .
```


View
Move to
Copy to
Delete
Rename
New folder
Properties
Open
Edit
History
Select all
Select none
Invert selection
Organize
New
Open
Select
S > source > repos > Project2 > Project2
Search Project2
Name
Date modified
Type
Size
Debug
7/8/2018 10:41 PM
File folder
Daily-Sales-Report-7-8-2018.txt
7/8/2018 11:02 PM
Text Document
1 KB
Project2.vcxproj
7/8/2018 9:24 PM
VC++ Project
6 KB
Project2.vcxproj.filters
7/8/2018 9:24 PM
VC++ Project Filte...
1 KB
Project2.vcxproj.user
5/8/2018 10:59 AM
Per-User Project O...
1 KB
7 bytes
State: Shared
Daily-Sales-Report-7-8-2018.txt - Notepad
File Edit Format View Help
----Daily Sales Report 7-8-2018----
Total customers for today: 0
Combo Quantity Sold Sales Amount
A 0 0.00
B 0 0.00
C 0 0.00
D 0 0.00
TOTALS 0 0.00
TOTAL SST charged = 0.00
TOTAL RM collected = 0.00
----REPORT END----

Run 3

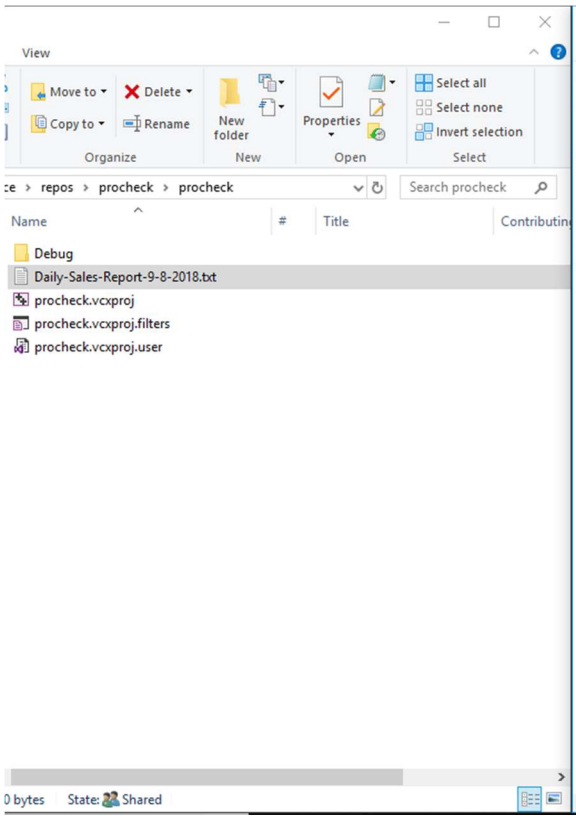
Run 3 scenario: More than 1 customer, 1 customer ordered a total of 2 sets of all combos,
Customer decided to cancel all combo A order (cashier enter negative quantity)
Customer pay insufficient money twice, report generated to file (Delete run 2 report first)
(Sales Report should automatically be displayed after last customer)

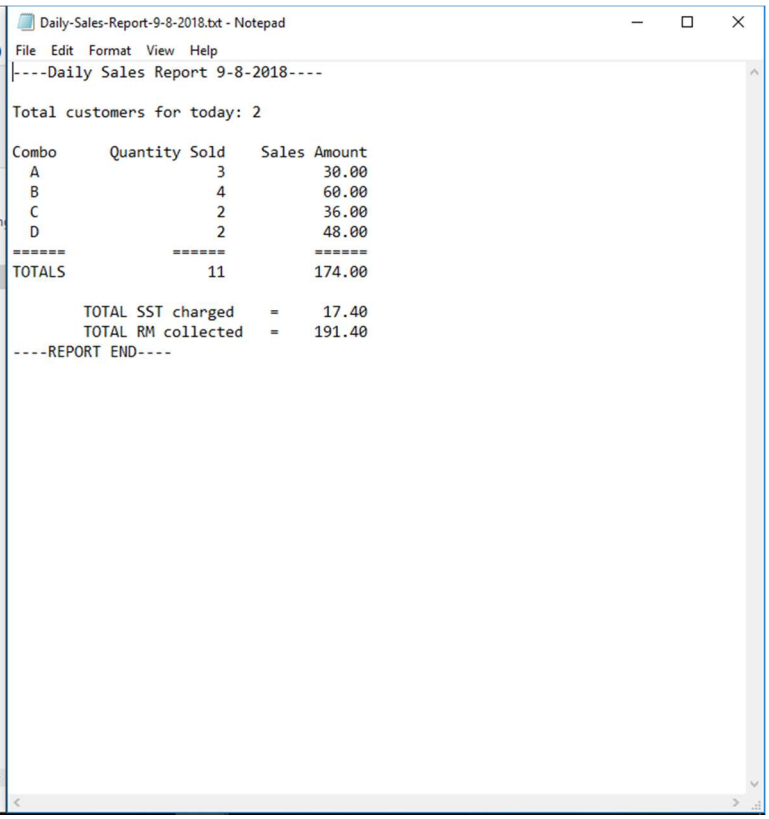
Inputs						**Expected Results / Outputs**			
Cust #	Combo ordered	Qty	Payment	Cancel?	Write to file?	Combo charges	SST 10%	Total charges	Change to be given
1	A	2				2 * 10 = 20	11.40	125.40	
	B	2	Pay: 100			2 * 15 = 30			(Prompt)
	C	2	Pay: 120			2 * 18 = 36 (114)			(Prompt)
	D	2	Pay: 126			2 * 24 = 48			0.60
	A	-2				2 * 10 = 20			
2	A	3	100			3 * 10 = 30 (60)	6	66	34
	B	2				2 * 15 = 30			
					Yes				
						Total Customers: 1 Total Combo Sales: A 3 * 10 = 30 B 4 * 15 = 60 C 2 * 18 = 36 D 2 * 24 = 48	17.40	191.40	
						11 174			

Daily-Sales-Report-DD-MM-YYYY.txt created in present working directory

```
 ____  ____  ___   ____  _  _  ____  ____  ____  _  _   __  ____
/ ___)(  _ \/ __) (  _ \/ )( \(  _ \(  _ \(  _ \( \/ ) /  \(  _ \
\___ \ ) _ ( (__   ) _ ( \/ \/ )   / ) _ ( ) _ (/ \/ \(  O )) D (
(____/(____/\___) (____/\_)(_/(__\_)(____/(____/\_)(_/ \__/(____/

You are customer: 1

----MENU----
Combo Contents                                                                                                  Price (RM)

A     Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola Drink (L)                                  10.00
B     Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink (L)                                        15.00
C     2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)                                                         18.00
D     3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2 Pepsi drink (S)                             24.00

----ORDERING----
Combo A,B,C,D (other = exit, X = cancel): A
Quantity        : 2
        Combo A x 2 @ RM    10.00 = RM     20.00
Combo A,B,C,D (other = exit, X = cancel): B
Quantity        : 2
        Combo B x 2 @ RM    15.00 = RM     30.00
Combo A,B,C,D (other = exit, X = cancel): C
Quantity        : 2
        Combo C x 2 @ RM    18.00 = RM     36.00
Combo A,B,C,D (other = exit, X = cancel): D
Quantity        : 2
        Combo D x 2 @ RM    24.00 = RM     48.00
Combo A,B,C,D (other = exit, X = cancel): A
Quantity        : -2
        Combo A x -2 @ RM    10.00 = RM    -20.00
Combo A,B,C,D (other = exit, X = cancel): 100

----BILLING----
        Gross total     = RM   114.00
        TOTAL PAYABLE   = RM   125.40
        Amount Paid (<= -999 = cancel)  = RM    100

        Insufficient funds, minimum payable is: RM 125.40

        Amount Paid (<= -999 = cancel)  = RM    120

        Insufficient funds, minimum payable is: RM 125.40

        Amount Paid (<= -999 = cancel)  = RM    126
        Change          = RM    0.60

Thank you, have a nice day and see you later!

Next customer? (y/n): y
You are customer: 2

----MENU----
Combo Contents                                                                                                  Price (RM)

A     Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola Drink (L)                                  10.00
B     Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink (L)                                        15.00
C     2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)                                                         18.00
D     3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2 Pepsi drink (S)                             24.00

----ORDERING----
Combo A,B,C,D (other = exit, X = cancel): A
Quantity        : 3
```

```
        Combo A x 3 @ RM    10.00 = RM     30.00
Combo A,B,C,D (other = exit, X = cancel): B
Quantity        : 2
        Combo B x 2 @ RM    15.00 = RM     30.00
Combo A,B,C,D (other = exit, X = cancel): u

----BILLING----
        Gross total     = RM    60.00
        TOTAL PAYABLE   = RM    66.00
        Amount Paid (<= -999 = cancel)  = RM    100
        Change          = RM    34.00

Thank you, have a nice day and see you later!

Next customer? (y/n): n

----DAILY SALES REPORT----
Total customers for today: 2

Combo Sales For Today

Combo      Quantity Sold    Sales Amount
  A                    3           30.00
  B                    4           60.00
  C                    2           36.00
  D                    2           48.00

======            ======          ======
TOTALS                11          174.00

        TOTAL SST charges     =     17.40
        TOTAL RM collected    =    191.40

Do you want to write the daily sales report to a file? (y/n): y
Successfully wrote to: .\Daily-Sales-Report-9-8-2018.txt

----ENDING----
~Thank you for using SFC Fast Food's Fast Food Billing System (F2BS)~
Made by: Hiew LS
--------------
Press any key to continue . . .
```





Constants & variables

```
//constants
#define PRICE_A 10.00
#define PRICE_B 15.00
#define PRICE_C 18.00
#define PRICE_D 24.00
#define TAX 10

//variables
int comboA, comboB, comboC, comboD, customers; // variables to track sales for daily sales report
int nowA, nowB, nowC, nowD; // keeps track of what the present customer order to make cancellation of
order possible.
double gross, sst, net;
report *dailyReport = malloc(sizeof(report));
double gTotal;
char nextCust, c;
char combo, c;
int qty, *ptr,  *nowPtr;
double price, cTotal, gTotal = 0;
char c;
double change, netTotal, paid, sstTotal;
char saveFile[35];
double totalA, totalB, totalC, totalD;
FILE *fp;
int totalQty;
struct tm *timeInfo;
time_t rawTime;
```

Constants			
Constant	Value	Data Type	Purpose
PRICE_A	10	double	Store price for combo A
PRICE_B	15	double	Store price for combo B
PRICE_C	18	double	Store price for combo C
PRICE_D	24	double	Store price for combo D
TAX	10	Int	Store tax/SST amount

Variables		
Variable	Data Type	Purpose
(*report* data type)		Q&A Do I need to list custom data types?
comboA	int	Store total daily order quantity for combo A
comboB	int	Store total daily order quantity for combo B
comboC	int	Store total daily order quantity for combo C
comboD	int	Store total daily order quantity for combo D
nowA	int	Store current customer order quantity for combo A
nowB	int	Store current customer order quantity for combo B
nowC	int	Store current customer order quantityfor combo C
nowD	int	Store current customer order quantity for combo D
gross	double	Store total gross sales (before tax) for the day
sst	double	Store total SST charged for the day
net	double	Store total net sales (after tax) for the day
(in *main* function)		
dailyReport	report*	Stores daily report details
gTotal	double	Store gross total sales (before tax) for current customer
nextCust	char	To indifcate whether there are anymore customer for the day
c	char	To receive unused characters to clear input buffer
(in *takeOrders* function)		
combo	char	Store combo type selected by customer
c	char	To receive unused characters to clear input buffer
qty	int	Store quantity for amount of current combo ordered
ptr	int*	Allow updating of total combo sales quantity in report
nowPtr	int*	Alllow updating of current customer total combo sales quantity in report
price	double	Store price of combo to be ordered
cTotal	double	Store current customer total combo sales amount (in RM)
gTotal	double	Store current customer gross total sales amount (in RM)
(in *billing* function)		
c	char	To receive unused characters to clear input buffer
change	double	Store change to be given back to customer after paying
netTotal	double	Store net total payable by customer
paid	double	Store amount paid by customer
sstTotal	double	Store amount of SST payable by customer
(in *statement* function)		
saveFile[35]	char	Store file name for saving daily sales report
totalA	double	Store total amount (RM) collected for combo set A
totalB	double	Store total amount (RM) collected for combo set B
totalC	double	Store total amount (RM) collected for combo set C
totalD	double	Store total amount (RM) collected for combo set B
fp	FILE*	To write daily sales report to a file
totalQty	int	To store total quantity of combo sold in a day
timeInfo	struct tm	To store current local time information
rawTime	time_t	To store current system time raw values

Program Listing

c:\Users\User-i300\Downloads\hiewLongShun.c 1

```c
/*****************************************************************************

Name: hiewLongShun.c
Version: 1.2
Description: A program to enable the restaurant to bill their customers quickly,
             and at the end of the day produce a Daily Sales Report.

*****************************************************************************/
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#pragma warning(disable:4996)

//constants
#define PRICE_A 10.00
#define PRICE_B 15.00
#define PRICE_C 18.00
#define PRICE_D 24.00
#define TAX 10

//custom data types
typedef struct
{
    int comboA, comboB, comboC, comboD, customers; // variables to track sales for daily sales report
    int nowA, nowB, nowC, nowD; // keeps track of what the present customer order to make cancellation of
      order possible.
    double gross, sst, net;

}
report;

//function prototypes
void printMenu();
double takeOrders(report *salesReport);
int billing(double *grossTotal, report *salesReport);
int statement(report *daily);
int cancel(report *report);

int main()
{
    //variables
    report *dailyReport = malloc(sizeof(report));
    double gTotal;
    char nextCust, c;

    //display information
    //display splash screen
    printf(" ____  ____  ___    ____  __  ____  ____  ____  __    __  ____ \n"
        "/ ___)(  __)/ __)  (  __)/ _\\ / ___)(_  _)(  __)/  \\  /  \\(    \\\n"
        "\\___ \\ ) _)( (__    ) _)/    \\\\___ \\  )(   ) _)(  O )(  O )) D (\n"
        "(____/(__)  \\___)  (__) \\_/\\_/(____/ (__) (__)  \\__/  \\__/(____/\n\n");

    //set all elements of dailyReport to 0.
    dailyReport->customers = 0;
    dailyReport->comboA = 0;
    dailyReport->comboB = 0;
    dailyReport->comboC = 0;
    dailyReport->comboD = 0;
    dailyReport->gross = 0;
    dailyReport->sst = 0;
    dailyReport->net = 0;

```

```c
    //TODO: Logo
    do
    {
        //Print customer number
        dailyReport->customers += 1;
        printf("You are customer: %d\n\n", dailyReport->customers);

        //display menu
        printMenu();

        //take orders
        gTotal = takeOrders(dailyReport);

        //process billing
        billing(&gTotal, dailyReport);

        //display ending screen
        printf("\nThank you, have a nice day and see you later!\n\n");

        //ask for next customer
        printf("Next customer? (y/n): ");
        nextCust = getchar();
        while ((c = getchar()) != '\n' && c != EOF);
    }
    while(tolower(nextCust) == 'y');

    //print Report
    statement(dailyReport);

    //Display ending message
    printf("\n" "----ENDING----\n");
    printf("~Thank you for using SFC Fast Food's Fast Food Billing System (F2BS)~\n");
    printf("Made by: Hiew LS\n");
    printf("--------------\n");
    system("pause");
    free(dailyReport);
    return 0;
}

void printMenu() //prints menu
{
    printf("----MENU----\n");
    printf("%-5s %-100s %s\n\n", "Combo", "Contents", "Price (RM)");
    printf("%-5s %-100s %10.2f\n", "A", "Chicken Burger + Potato Wedges (S) +  6 Chicken Nuggets + Coca-cola
      Drink (L)", PRICE_A);
    printf("%-5s %-100s %10.2f\n", "B", "Marinated Fried Chicken + Chicken Rice + Coleslaw (L) + Pepsi drink
      (L)", PRICE_B);
    printf("%-5s %-100s %10.2f\n", "C", "2 BBQ Fried Chicken + Chicken Rice + 2 Pepsi drink (S)", PRICE_C);
    printf("%-5s %-100s %10.2f\n", "D", "3 Marinated Fried Chicken + 2 Chicken Rice + 6 Chicken Nuggets + 2
      Pepsi drink (S)", PRICE_D);
}

double takeOrders(report *salesReport)
{
    //variables
    char combo, c;
    int qty, *ptr,  *nowPtr;
    double price, cTotal, gTotal = 0;

    //initialize now[Combo] in report to 0 - keep track of current order
    salesReport->nowA = 0;
    salesReport->nowB = 0;
    salesReport->nowC = 0;
    salesReport->nowD = 0;
```

```c

    //let cashier know ready to order
    putchar('\n');
    printf("----ORDERING----\n");

    //get combo type
    while(1)
    {
        //get combo type
        printf("Combo A,B,C,D (other = exit, X = cancel): ");
        combo = getchar();
        while ((c = getchar()) != '\n' && c != EOF);

        //set price - nowPtr is used as duplicate entry to track order cancellation
        switch (combo)
        {
            case 'A':
            {
                price = PRICE_A;
                ptr = &(salesReport->comboA);
                nowPtr = &(salesReport->nowA);
                break;
            }
            case 'B':
            {
                price = PRICE_B;
                ptr = &(salesReport->comboB);
                nowPtr = &(salesReport->nowB);
                break;
            }
            case 'C':
            {
                price = PRICE_C;
                ptr = &(salesReport->comboC);
                nowPtr = &(salesReport->nowC);
                break;
            }
            case 'D':
            {
                price = PRICE_D;
                ptr = &(salesReport->comboD);
                nowPtr = &(salesReport->nowD);
                break;
            }
            case 'X':
            {
                return cancel(salesReport);
            }
            default:
            {
                return(gTotal);
            }
        }

        //get quantity
        printf("Quantity\t: ");
        scanf("%d", &qty);

        //add quantity to pointed combo in daily report
        *nowPtr = *ptr += qty;
        while ((c = getchar()) != '\n' && c != EOF);

        //calculate combo total
        cTotal = qty * price;
```

```
189
190         //display combo total
191         printf("\tCombo %c x %d @ RM %8.2f = RM %9.2f\n", combo, qty, price, cTotal);
192
193         //add combo total to gross total
194         gTotal += cTotal;
195     }
196 }
197
198 int billing(double *grossTotal, report *salesReport)
199 {
200     //variables
201     char c;
202     double change, netTotal, paid, sstTotal;
203
204     //check if gross is acceptably equal zero
205     if (*grossTotal < 0.0001)
206     {
207         return 0; // no need to calculate, since order is empty
208     }
209
210     putchar('\n');
211     printf("----BILLING----\n");
212
213     //calculate SST amount
214     sstTotal = *grossTotal * TAX / 100;
215
216     //calculate net total
217     netTotal = *grossTotal + sstTotal;
218
219     //display gross total
220     printf("\tGross total\t= RM %8.2f\n", *grossTotal);
221
222     //display net total
223     printf("\tTOTAL PAYABLE\t= RM %8.2f\n", netTotal);
224
225     //accept payment
226     printf("\tAmount Paid (<= -999 = cancel)\t= RM    ");
227     scanf("%lf", &paid);
228     while ((c = getchar()) != '\n' && c != EOF);
229
230     //check if cancel order
231     if (paid < -998)
232     {
233         return cancel(salesReport);
234     }
235
236     //check if insufficient funds
237     while (paid < netTotal)
238     {
239         //accept payment
240         printf("\n" "\tInsufficient funds, minimum payable is: RM %.2f\n\n", netTotal);
241         printf("\tAmount Paid (<= -999 = cancel)\t= RM    ");
242         scanf("%lf", &paid);
243         while ((c = getchar()) != '\n' && c != EOF);
244
245         //check if cancel order
246         if (paid < -998)
247         {
248             return cancel(salesReport);
249         }
250     }
251
252     //calculate change
```

```
253     change = paid - netTotal;
254     //display change
255     printf("\tChange\t\t= RM\t%1.2f\n", change);
256
257     //accumulate gross total
258     salesReport->gross += *grossTotal;
259
260     //accumulate tax total
261     salesReport->sst += sstTotal;
262
263     //accumulate net total
264     salesReport->net += netTotal;
265
266     return 0;
267 }
268
269 int statement(report *daily)
270 {
271     //variables
272     char saveFile[35];
273     double totalA, totalB, totalC, totalD;
274     FILE *fp;
275     int totalQty;
276     struct tm *timeInfo;
277     time_t rawTime;
278
279     //process
280
281     //calculate combo sales amount
282     totalA = daily->comboA * PRICE_A;
283     totalB = daily->comboB * PRICE_B;
284     totalC = daily->comboC * PRICE_C;
285     totalD = daily->comboD * PRICE_D;
286
287     //calculate total combo sold
288     totalQty = daily->comboA + daily->comboB + daily->comboC + daily->comboD;
289
290     //output
291
292     //let cashier know report generated
293     printf("\n" "----DAILY SALES REPORT----\n");
294
295     //print total customers
296     printf("Total customers for today: %d\n\n", daily->customers);
297
298     //display combo sales
299     printf("Combo Sales For Today\n\n");
300     printf("%-10s %-13s %15s\n", "Combo", "Quantity Sold", "Sales Amount");
301     printf("  %-8c %13d %15.2f\n", 'A', daily->comboA, totalA);
302     printf("  %-8c %13d %15.2f\n", 'B', daily->comboB, totalB);
303     printf("  %-8c %13d %15.2f\n", 'C', daily->comboC, totalC);
304     printf("  %-8c %13d %15.2f\n", 'D', daily->comboD, totalD);
305
306     //display gross total
307     putchar('\n');
308     printf("%-10s %13s %15s\n", "======", "======", "======");
309     printf("%-10s %13d %15.2f\n\n", "TOTALS", totalQty, daily->gross);
310
311     printf("\t%-20s = %9.2f\n", "TOTAL SST charges", daily->sst);   //print total SST collected
312     printf("\t%-20s = %9.2f\n", "TOTAL RM collected", daily->net);  //print net total
313
314     printf("\n" "Do you want to write the daily sales report to a file? (y/n): ");  //ask user if want to
        write to file
315     if (tolower(getchar()) == 'y')
```

```
316     {
317         //get file name in format "Daily-Sales-Report-DD-MM-YYYY"
318         time(&rawTime);
319         timeInfo = localtime(&rawTime);
320         snprintf(saveFile, 35, "Daily-Sales-Report-%d-%d-%d.txt", timeInfo->tm_mday, timeInfo->tm_mon + 1,
              timeInfo->tm_year + 1900);
321
322         //create/open the save file file
323         fp = fopen(saveFile, "a");
324         if (fp == NULL)
325         {
326             printf("File cannot be created, please copy-paste output into a new text file.");
327             return 1;
328         }
329
330         fprintf(fp, "----Daily Sales Report %d-%d-%d----\n\n", timeInfo->tm_mday, timeInfo->tm_mon + 1,
              timeInfo->tm_year + 1900);
331
332         //print total customers
333         fprintf(fp, "Total customers for today: %d\n\n", daily->customers);
334
335         //print combo sales details
336         fprintf(fp, "%-10s %-13s %15s\n", "Combo", "Quantity Sold", "Sales Amount");
337         fprintf(fp, "  %-8c %13d %15.2f\n", 'A', daily->comboA, totalA);
338         fprintf(fp, "  %-8c %13d %15.2f\n", 'B', daily->comboB, totalB);
339         fprintf(fp, "  %-8c %13d %15.2f\n", 'C', daily->comboC, totalC);
340         fprintf(fp, "  %-8c %13d %15.2f\n", 'D', daily->comboD, totalD);
341
342         fprintf(fp, "%-10s %13s %15s\n"
343                     "%-10s %13d %15.2f\n\n", "======", "======", "======", "TOTALS", totalQty, daily->gross);
344
345         //print total SST collected
346         fprintf(fp, "\t%-20s = %9.2f\n"
347                     "\t%-20s = %9.2f\n", "TOTAL SST charged", daily->sst, "TOTAL RM collected", daily->net);
348
349         fprintf(fp, "----REPORT END----");
350
351         printf("Successfully wrote to: .\\Daily-Sales-Report-%d-%d-%d.txt\n", timeInfo->tm_mday, timeInfo-
              >tm_mon + 1, timeInfo->tm_year + 1900);
352         fclose(fp);
353     }
354     return 0;
355 }
356
357 //cancel order function
358
359 int cancel(report *report)
360 {
361     printf("\n" "----Cancellation----\n\n");
362
363     //deduct combo's cancelled for current order
364     report->comboA -= report->nowA;
365     report->comboB -= report->nowB;
366     report->comboC -= report->nowC;
367     report->comboD -= report->nowD;
368
369     //subtract customer count
370     report->customers--;
371
372     //inform cashier of cancellation
373     puts("Order cancelled, customer count - 1.\n");
374
375     return 0;
376 }
377
378
```