

**CSC138 – Structured Programming**

**GROUP: JCS1102E**

**PROJECT TITLE: FUNBUN SYSTEM**

**Prepared By:**

|  |  |
| --- | --- |
| **Name** | **Matrix ID** |
| MUHAMMAD AQIL IRFAN BIN BADRUL HISHAM | 2017258974 |
| MUHAMMAD HAFIEZUDIN BIN ROSDI | 2017224744 |
| MUHAMMAD RAMZI AMRI BIN ROZMI | 2017231946 |

**Prepared For**

Nafisah Binti Amin

**Submission Date**

6th June 2018

Report Contents Page

[**1.0 INTRODUCTION 2**](#_Toc515914891)

[**2.0 OBJECTIVES 3**](#_Toc515914892)

[**3.0 FULL PROGRAM 4**](#_Toc515914893)

[**4.0 SAMPLE INPUT OUTPUT 16**](#_Toc515914894)

[**4.1 PERSONAL DATA 24**](#_Toc515914895)

[**4.2 DATA SEARCHED 24**](#_Toc515914896)

[**4.3 LIST CUSTOMER DATA 25**](#_Toc515914897)

# 1.0 INTRODUCTION

As for this project, our team decided to make a bakery system named *Funbun* system which only consist two types of buns which are red beans and durian flavour. We decided to choose those flavours because we think that is the most unique flavoured so the customers would like to buy it. We sell the buns in two ways which are half dozen where it has six pieces of buns and a dozen which has 12 pieces of buns so the customers can choose which one they can buy. For the price of the red beans flavour, it will cost RM5.00 for a half dozen and Rm10.00 for dozen while the durian flavour will cost at RM6.00 for half dozen and RM 12.00 for a dozen. For the payment method, the customers can either pay through bank in or cash on delivery(COD). We will deliver the buns and as for the delivery, there are no delivery fees which means no additional fees that customers have to pay, they only have to pay for the buns only. For those customers who live near Segamat, they can use the COD method because we only provide this service in Segamat area only which is the customers will meet up with us at One Segamat Shopping Mall which is addressed at 1 Shopping Complex, Second Floor, Jalan Kolam Air, Kampung Gubah,85000, Segamat, Johor which is at Segamat.This service also don’t have any additional fees so the customers only have to pay the price for the buns only. So we hope when we created this system, people will be attracted by this system and consider buying the buns.

# 2.0 OBJECTIVES

The objectives that we want to achieve are divided into six objectives which are:

1. To count the number of customers that order the bun for each type of bun.
2. To calculate the total average of the income in a month for each bun.
3. To calculate total income that has received after a customer uses our system in a month.
4. To calculate the highest order and identify the highest sales in a month.
5. To calculate the lowest order and identify the lowest sales.
6. To sort out from highest to lowest on how much quantity of bread has been sold per month by each type of buns.
7. To search the customer details and update by each type of buns.

# 3.0 FULL PROGRAM

#include <iostream>

#include <conio.h>

#include <iomanip>

#include <cstring>

#include <fstream>

struct info{

char name[30];

char no[15];

char payment;

char address[70];

double price;

int tDurian;

int tRedBean;

char method[20];

};

using namespace std;

// All of these are function declaration

void types();

void Data(info[], ofstream&, int&, int);

void paymentMethod(info[], int);

void highestLowest(info[], int);

void sorting(info[], int);

void search(info[], int);

int main()

{

char answer;

ofstream buns;

info data[100]={ };

int count=0, j, i=0;

double Total=0.00;

buns.open("Personal Data.txt");

do

{

cout<<"Welcome to Funbun system"<<endl;

types(); //to display the type of buns as well as its price

data[i].tDurian=0;

data[i].tRedBean=0;

Data(data, buns, j, i); //to get data from user

cout<<"Final price: RM"<<data[i].price<<endl;

paymentMethod(data, i); //to inform the user about payment method

count++;

do

{

cout<<"Do you want to enter another customer? (Y/N): ";

cin>>answer;

}

while((answer!='Y' && answer!='y') && (answer!='N' && answer!='n'));

if(answer=='Y' || answer=='y')

{

i++;

}

cout<<endl;

}

while((answer=='Y' || answer=='y'));

for(int x=0; x<=j; x++)

{

Total=Total+data[x].price;

}

cout<<"Total customer: "<<count<<endl;

cout<<setprecision(2)<<fixed; //setprecision(2)<<fixed is to manipulate the output into 2 decimal places

cout<<"Total payment received : RM"<<Total<<endl;

cout<<"The average of income per customer is :RM"<<Total/count<<endl;

highestLowest(data, j); //to identify the highest and lowest order as well as the highest and lowest sales

sorting(data, j); //to sort from highest to lowest of quantity durian buns and red bean buns

buns.close();

search(data); //to search the name of customer

getch();

return 0;

}

void Data(info data[], ofstream& buns, int& j, int i) // Function definition must be used to call the calling module

{

int durian, redBean, tDurian=0, tRedBean=0;

char ans, method[20];

data[i].price=0;

cout<<"Enter customer name: ";

cin>>ws;

cin.getline(data[i].name,30);

cout<<"Enter phone number: ";

cin>>ws;

cin.getline(data[i].no,15);

do

{

cout<<"Enter payment method (A-Cash On Delivery / B-Delivery): ";

cin>>ws;

cin>>data[i].payment;

cout<<endl;

}

while((data[i].payment!='A' && data[i].payment!='a') && (data[i].payment!='B' && data[i].payment!='b'));

cout<<"If you do not want to buy any buns, please enter 0 and you have to enter either 6 or 12 only"<<endl;

do

{

cout<<"Please enter quantity Durian bun (6-Half dozen / 12-Dozen): ";

cin>>durian;

if(durian==6)

{

data[i].price=data[i].price+6;

}

else if(durian==12)

{

data[i].price=data[i].price+12;

}

else

{

durian=0;

data[i].price=data[i].price+0;

}

cout<<"Please enter quantity Red Bean bun (6-Half dozen / 12-Dozen): ";

cin>>redBean;

if(redBean==6)

{

data[i].price=data[i].price+5;

}

else if(redBean==12)

{

data[i].price=data[i].price+10;

}

else

{

redBean=0;

data[i].price=data[i].price+0;

}

data[i].tDurian+=durian;

data[i].tRedBean+=redBean;

cout<<"Total durian buns: "<<data[i].tDurian<<endl;

cout<<"Total red bean buns: "<<data[i].tRedBean<<endl;

cout<<"Your total buns: "<<data[i].tDurian+data[i].tRedBean<<endl<<endl;

do

{

cout<<"Do you want to order more (Y/N): ";

cin>>ans;

cout<<endl;

}

while((ans!='Y' && ans!='y') && (ans!='N' && ans!='n'));

}

while(ans=='Y' || ans=='y');

if(data[i].payment=='A' || data[i].payment=='a')

{

strcpy(data[i].method, "Cash on Delivery");

}

else

{

strcpy(data[i].method, "Delivery");

}

buns<<data[i].name<<"\* "<<data[i].no<<"\* "<<data[i].tDurian<<" "<<data[i].tRedBean<<" "<<data[i].method<<endl;

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

cout<<setw(23)<<left<<"Name"<<":"<<data[i].name<<endl;

cout<<setw(23)<<left<<"Phone no. "<<":"<<data[i].no<<endl;

cout<<setw(23)<<left<<"Quantity Durian bun "<<":"<<data[i].tDurian<<endl;

cout<<setw(23)<<left<<"Quatity Red Bean bun "<<":"<<data[i].tRedBean<<endl;

cout<<setw(23)<<left<<"Payment Method "<<":"<<data[i].method<<endl;

cout<<setprecision(2)<<fixed;

cout<<setw(23)<<left<<"Price "<<":RM"<<data[i].price<<endl;

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

cout<<"\n\tTHANK YOU FOR USING OUR SYSTEM"<<endl;

j=i;

}

void types()

{

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

cout<<"| Flavour | Quantity | Price (RM) |"<<endl;

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

cout<<"| Red Bean | 6 | 5.00 |"<<endl;

cout<<"| (A) | 12 | 10.00 |"<<endl;

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

cout<<"| Durian | 6 | 6.00 |"<<endl;

cout<<"| (B) | 12 | 12.00 |"<<endl;

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

}

void paymentMethod(info data[], int i)

{

char Name[30], information;

if(data[i].payment=='A' || data[i].payment=='a')

{

cout<<"Meet at Segamat"<<endl;

cout<<"1 Shopping Complex, First Floor, Jalan Kolam Air, Kampung Gubah, 85000 Segamat District, Johor, Malaysia"<<endl;

cout<<"From 9.00a.m.-11.00a.m"<<endl;

cout<<"For further information please contact 012-281 3154(Aqil)"<<endl<<endl;

}

else

{

cout<<"Please bank in at 12-256-10-045316-0 (Bank Islam) - Hafiezudin Rosdi"<<endl;

cout<<"Please type in your bank account name: ";

cin>>ws;

cin.getline(Name,30);

cout<<endl;

do

{

cout<<"Please enter your shipping address: "<<endl;

cin>>ws;

cin.getline(data[i].address,70);

cout<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Your name: "<<Name<<endl;

cout<<"Please check and confirm your address: "<<data[i].address<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Your information is correct (Y/N): ";

cin>>information;

cout<<endl;

}

while(information!='Y' && information!='y');

void highestLowest(info data[], int j)

{

int highestOrder=data[0].tDurian+data[0].tRedBean, highestSales=data[0].price;

int lowestOrder=data[0].tDurian+data[0].tRedBean, lowestSales=data[0].price;

for(int i=0; i<=j; i++)

{

if(highestOrder<(data[i].tDurian+data[i].tRedBean))

{

highestOrder=data[i].tDurian+data[i].tRedBean;

}

if(highestSales<data[i].price)

{

highestSales=data[i].price;

}

if(lowestOrder>(data[i].tDurian+data[i].tRedBean))

{

lowestOrder=data[i].tDurian+data[i].tRedBean;

}

if(lowestSales>data[i].price)

{

lowestSales=data[i].price;

}

}

cout<<"\nHighest order is: "<<highestOrder<<" buns"<<"\nHighest sales is: RM"<<highestSales<<endl;

cout<<"Lowest order is: "<<lowestOrder<<" buns"<<"\nLowest sales is: RM"<<lowestSales<<endl;

}

void sorting(info data[], int j)

{

int highestTdurian=data[0].tDurian, temp;

int highestTRedBean=data[0].tRedBean, Temp;

for(int i=0; i<=j; i++)

{

for(int z=0; z<=j-1; z++)

{

if(data[z].tDurian<data[z+1].tDurian)

{

temp=data[z].tDurian;

data[z].tDurian=data[z+1].tDurian;

data[z+1].tDurian=temp;

}

if(data[z].tRedBean<data[z+1].tRedBean)

{

Temp=data[z].tRedBean;

data[z].tRedBean=data[z+1].tRedBean;

data[z+1].tRedBean=Temp;

}

}

}

cout<<"\nHighest order to lowest order for Durian buns: "<<endl;

for(int i=0; i<=j; i++)

{

cout<<data[i].tDurian<<" ";

}

cout<<endl<<endl;

cout<<"Highest order to lowest order for Red Bean buns: "<<endl;

for(int i=0; i<=j; i++)

{

cout<<data[i].tRedBean<<" ";

}

cout<<endl;

}

void search(info data[], int j)

{

char input[30],code;

ofstream update;

update.open("Data searched.txt");

ifstream bun;

bun.open("Personal Data.txt");

ofstream personal;

personal.open("List customer data.txt");

int i=0;

while(!bun.eof() && (i<=j))

{

bun.getline(data[i].name, 30, '\*');

bun>>ws;

bun.getline(data[i].no, 15, '\*');

bun>>data[i].tDurian>>data[i].tRedBean;

bun>>ws;

bun.getline(data[i].method, 20);

personal<<setw(23)<<left<<"Name "<<":"<<data[i].name<<endl;

personal<<setw(23)<<left<<"Phone no. "<<":"<<data[i].no<<endl;

personal<<setw(23)<<left<<"Durian buns "<<":"<<data[i].tDurian<<endl;

personal<<setw(23)<<left<<"Red Bean buns "<<":"<<data[i].tRedBean<<endl;

personal<<setw(23)<<left<<"Method "<<":"<<data[i].method<<endl<<endl;

i++;

}

do

{

bool found = false;

cout<<"\nSearching name : ";

cin>>ws;

cin.getline(input,30);

int i=0;

for(int i=0; i<=j; i++)

{

if (strcmp(data[i].name,input)==0)

{

found = true;

cout<<"Data found"<<endl;

update<<setw(23)<<left<<"Name "<<":"<<data[i].name<<endl<<setw(23)<<left<<"Phone no. "<<":"<<data[i].no<<endl;

update<<setw(23)<<left<<"Durian buns "<<":"<<data[i].tDurian<<endl;

update<<setw(23)<<left<<"Red Bean buns "<<":"<<data[i].tRedBean<<endl;

update<<setw(23)<<left<<"Method "<<":"<<data[i].method<<endl<<endl;

}

}

if (!found)

cout << "The data is not exist!"<< endl;

do

{

cout<<"\nDo you want to continue serching (Y/N) : ";

cin>>ws;

cin>>code;

}

while((code!='Y' && code!='y') && (code!='N' && code!='n'));

i++;

}

while(code=='Y'||code=='y');

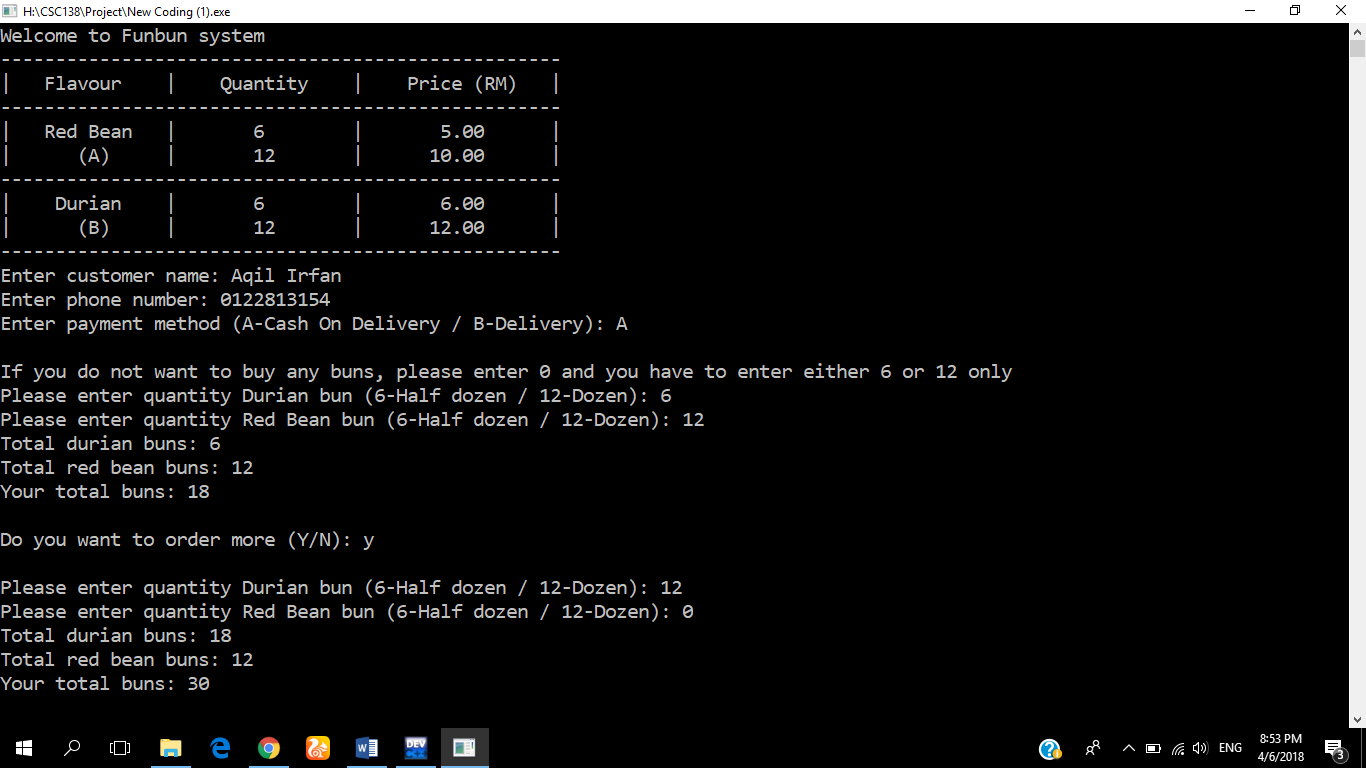
update.close();

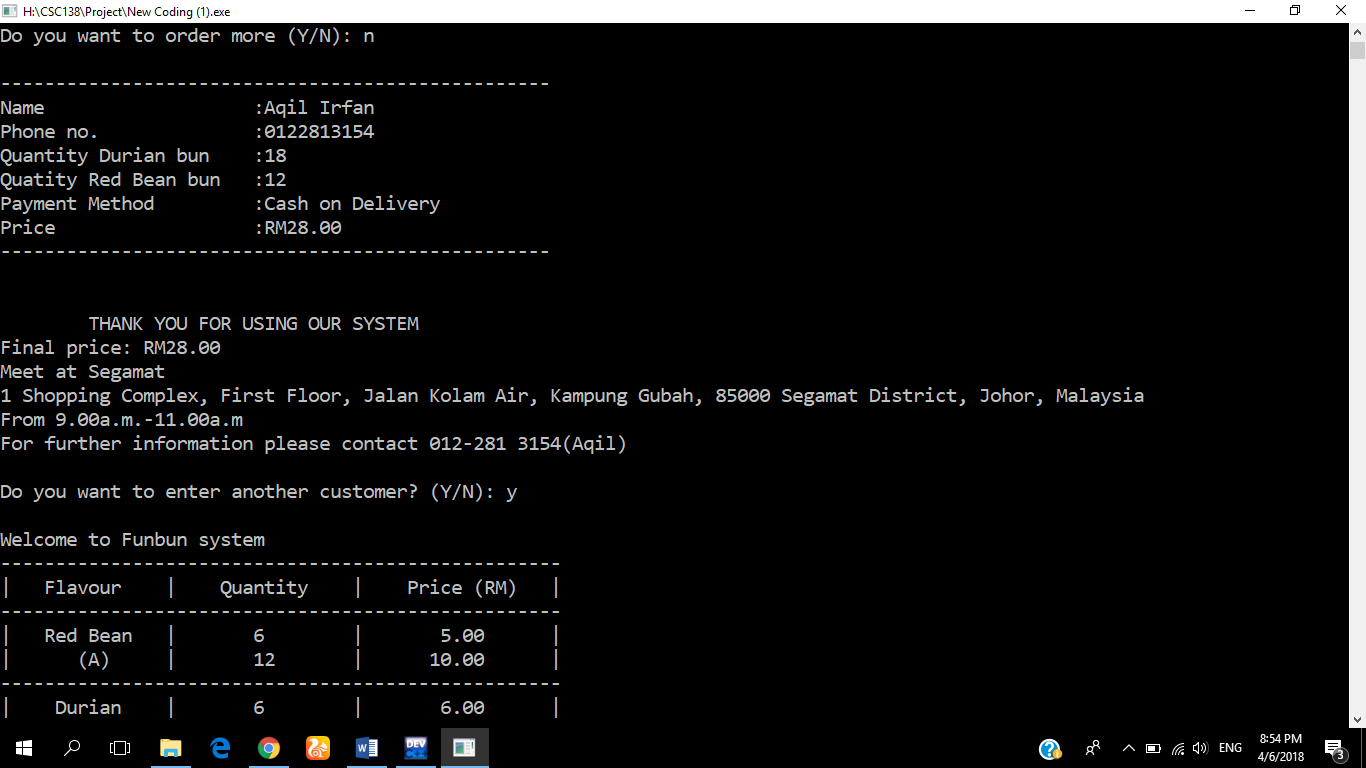
bun.close();

personal.close();

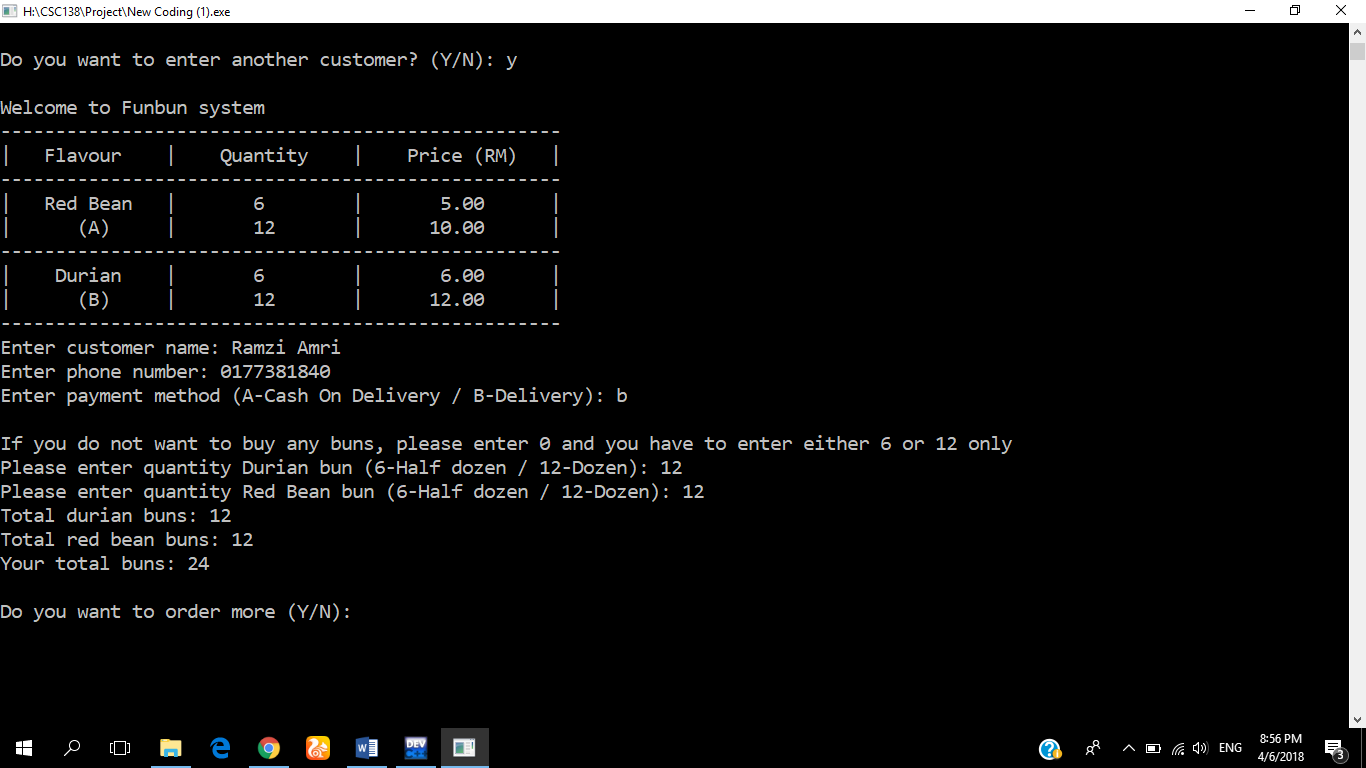
}

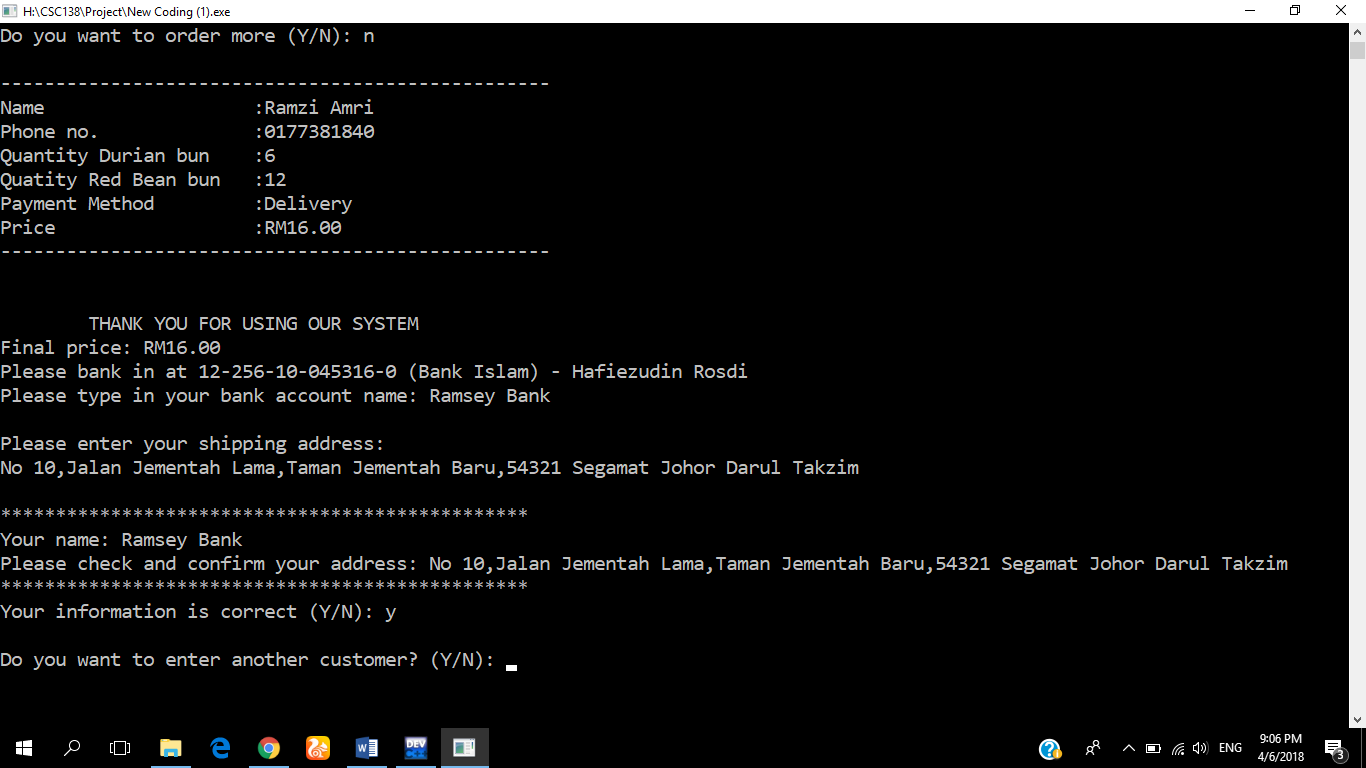
# 4.0 SAMPLE INPUT OUTPUT

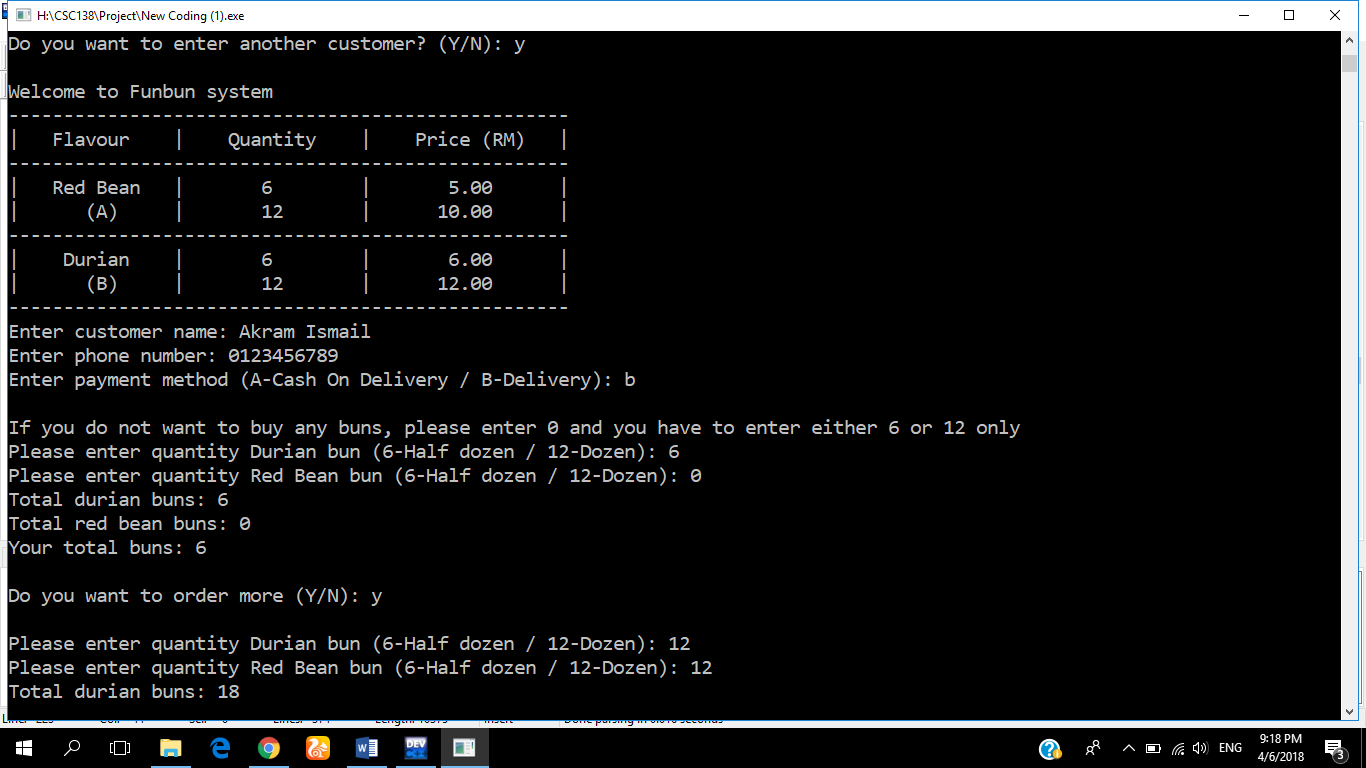


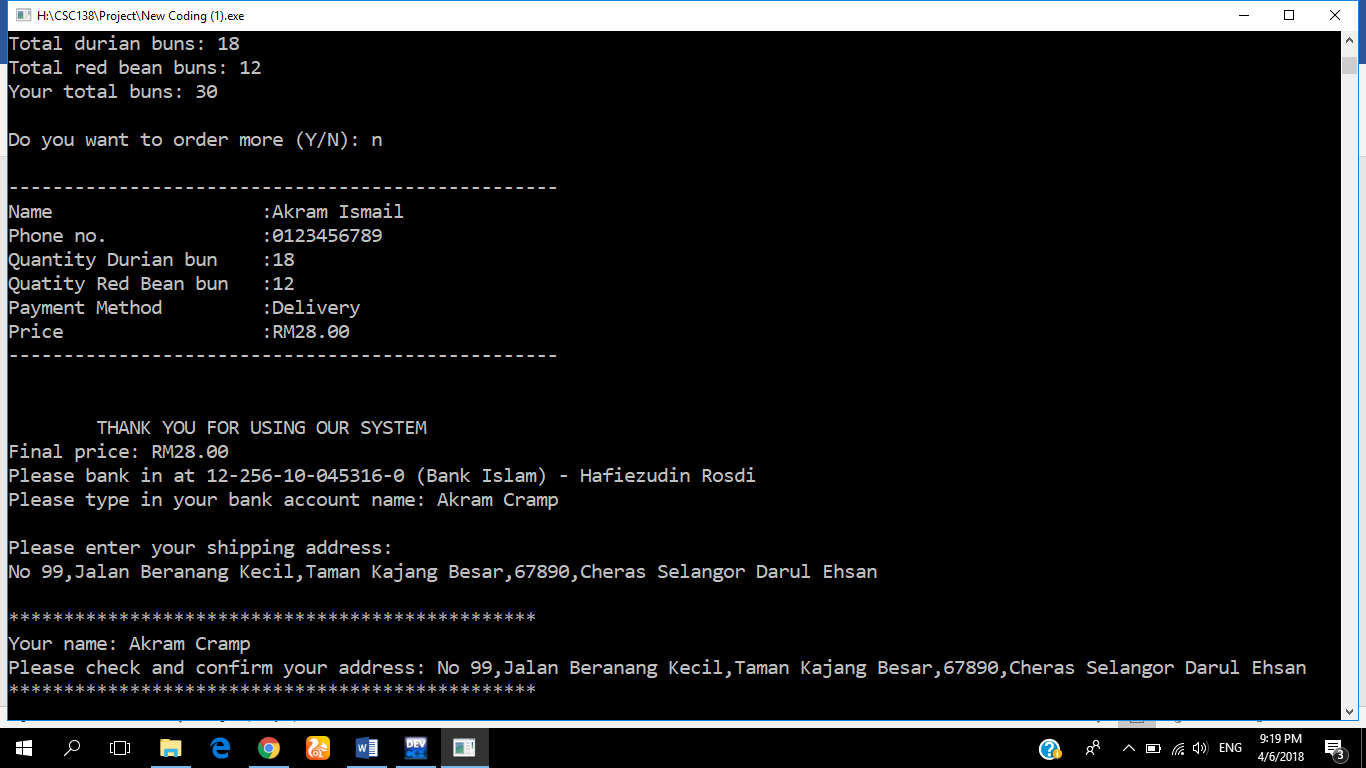


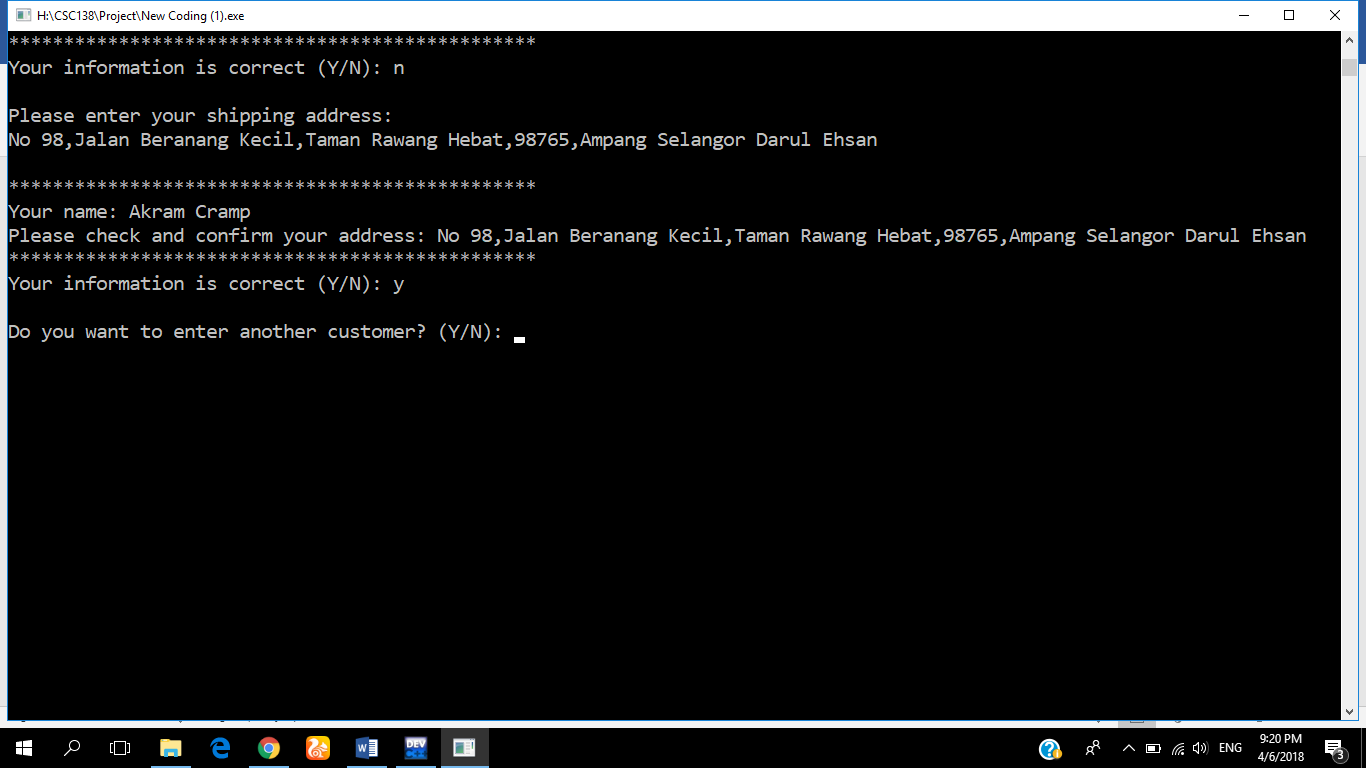


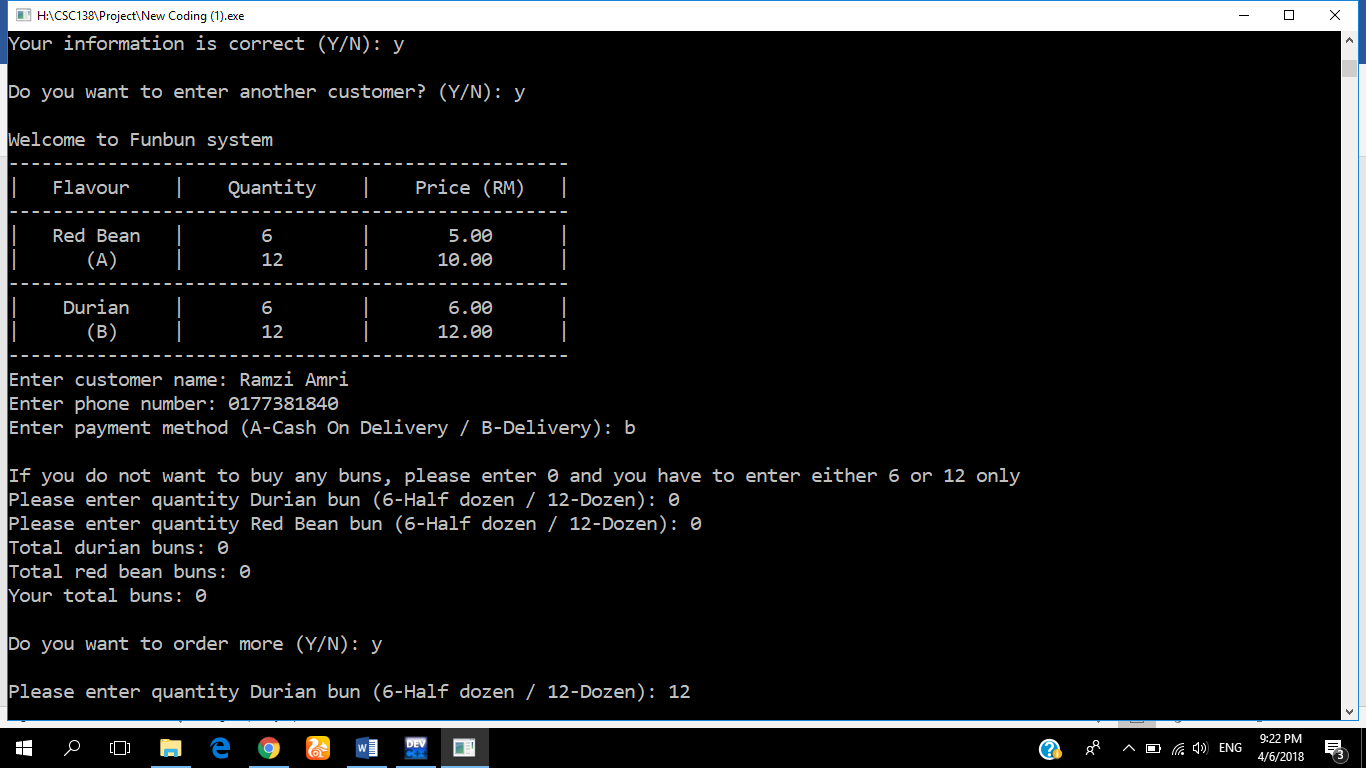


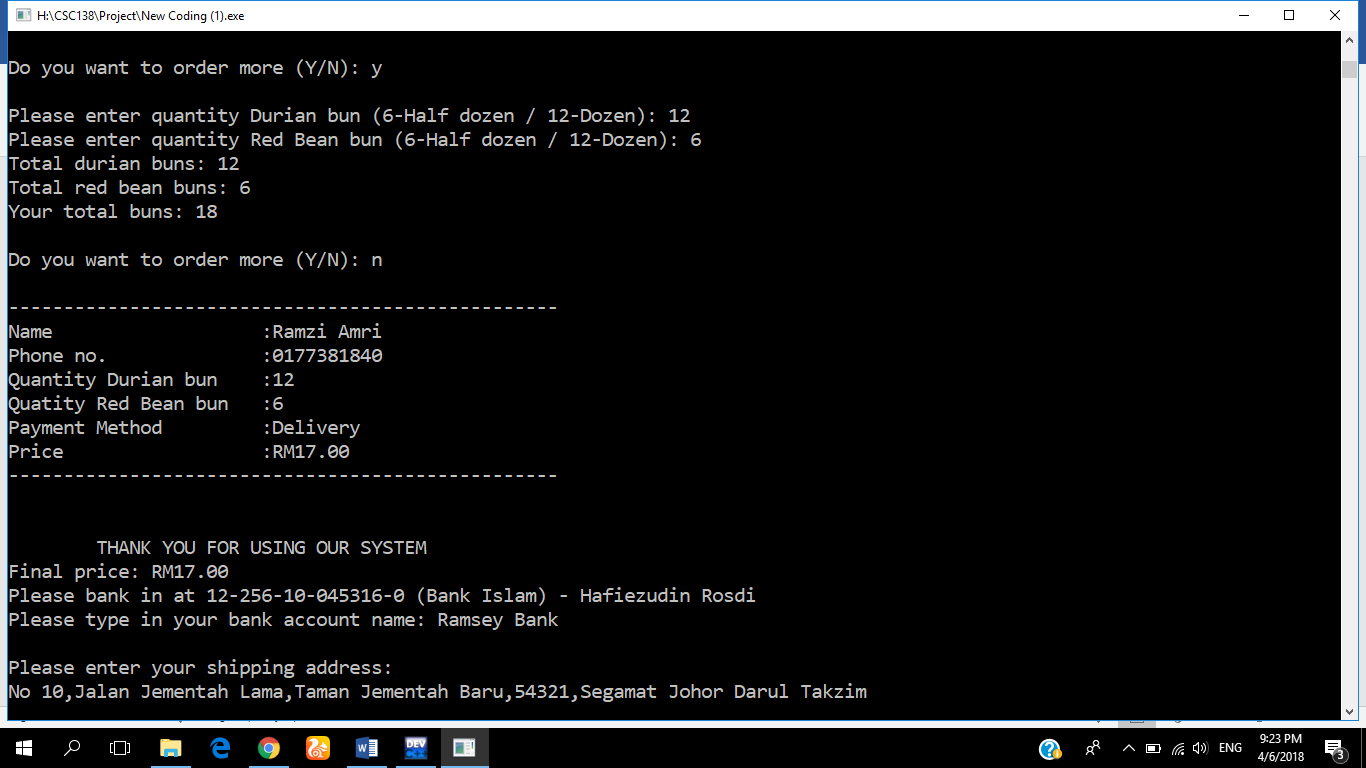


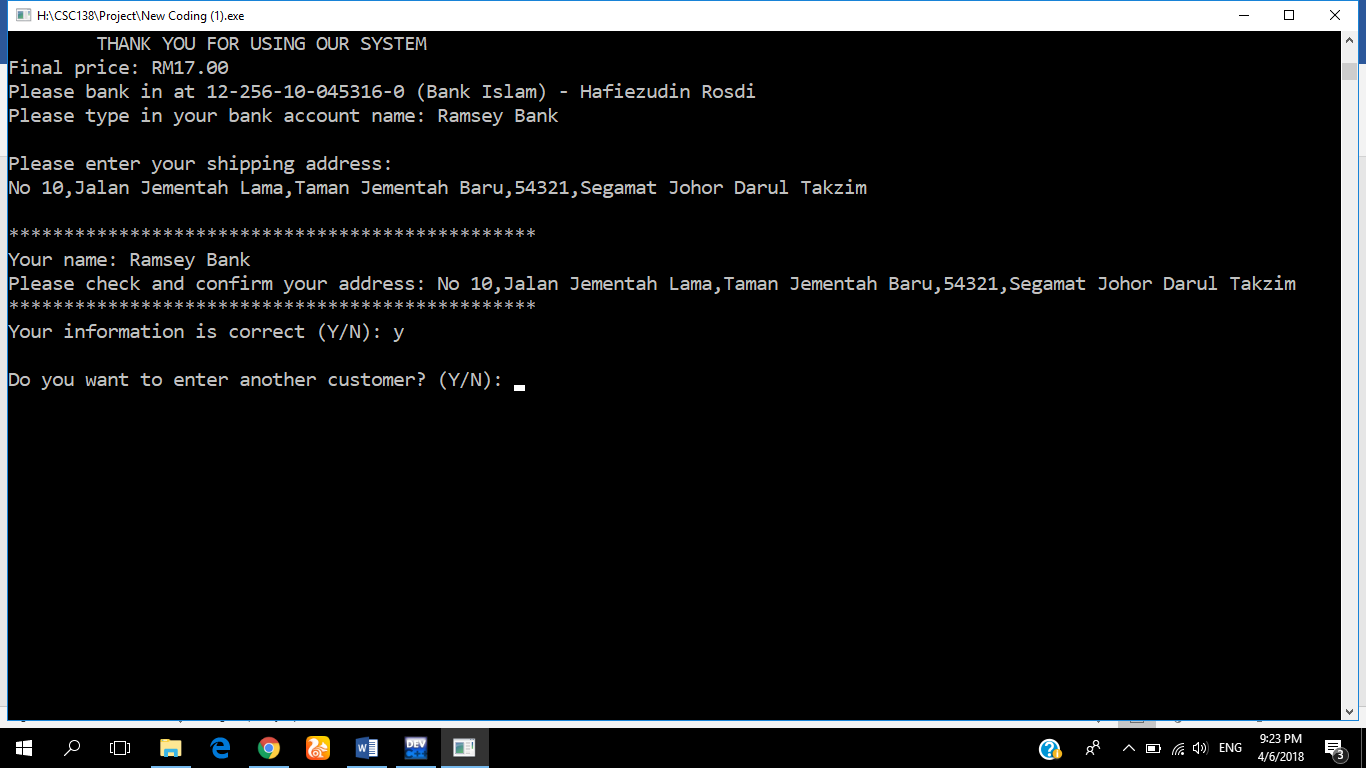


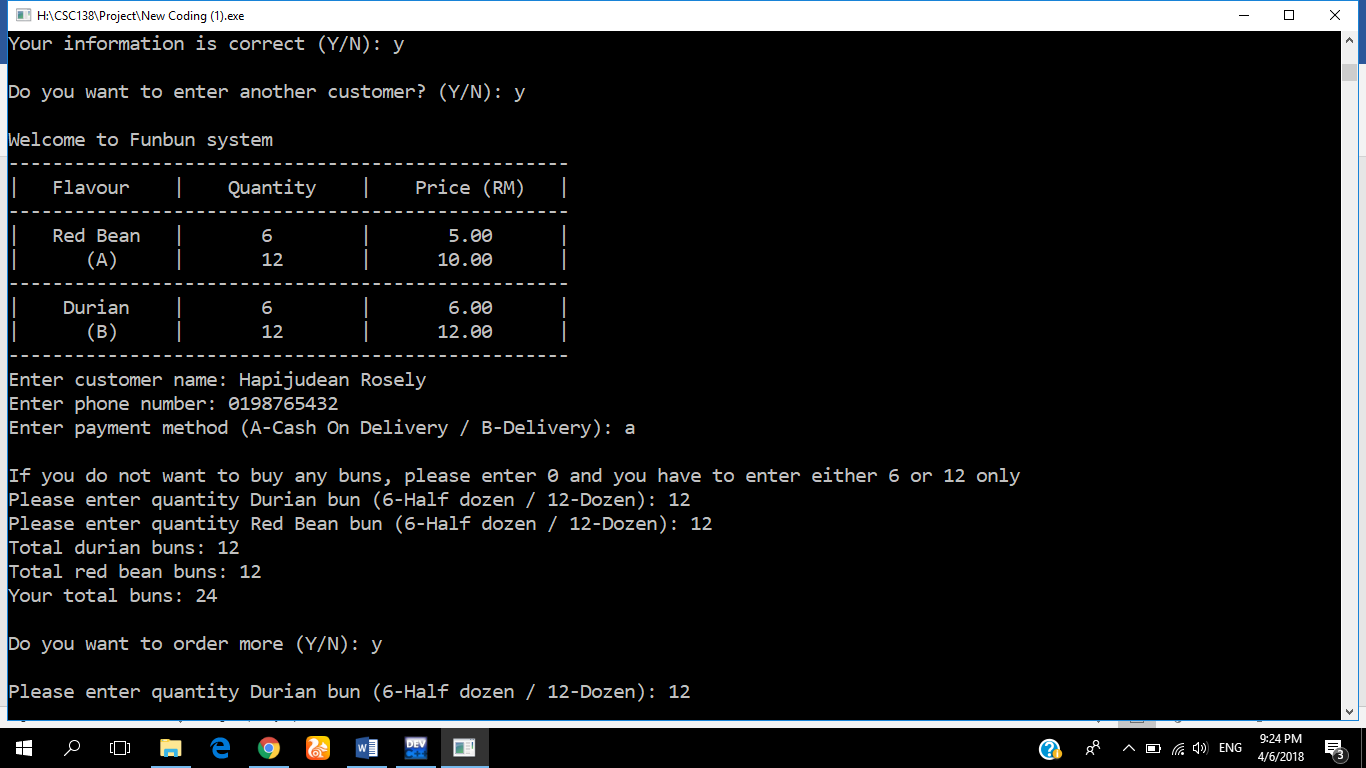


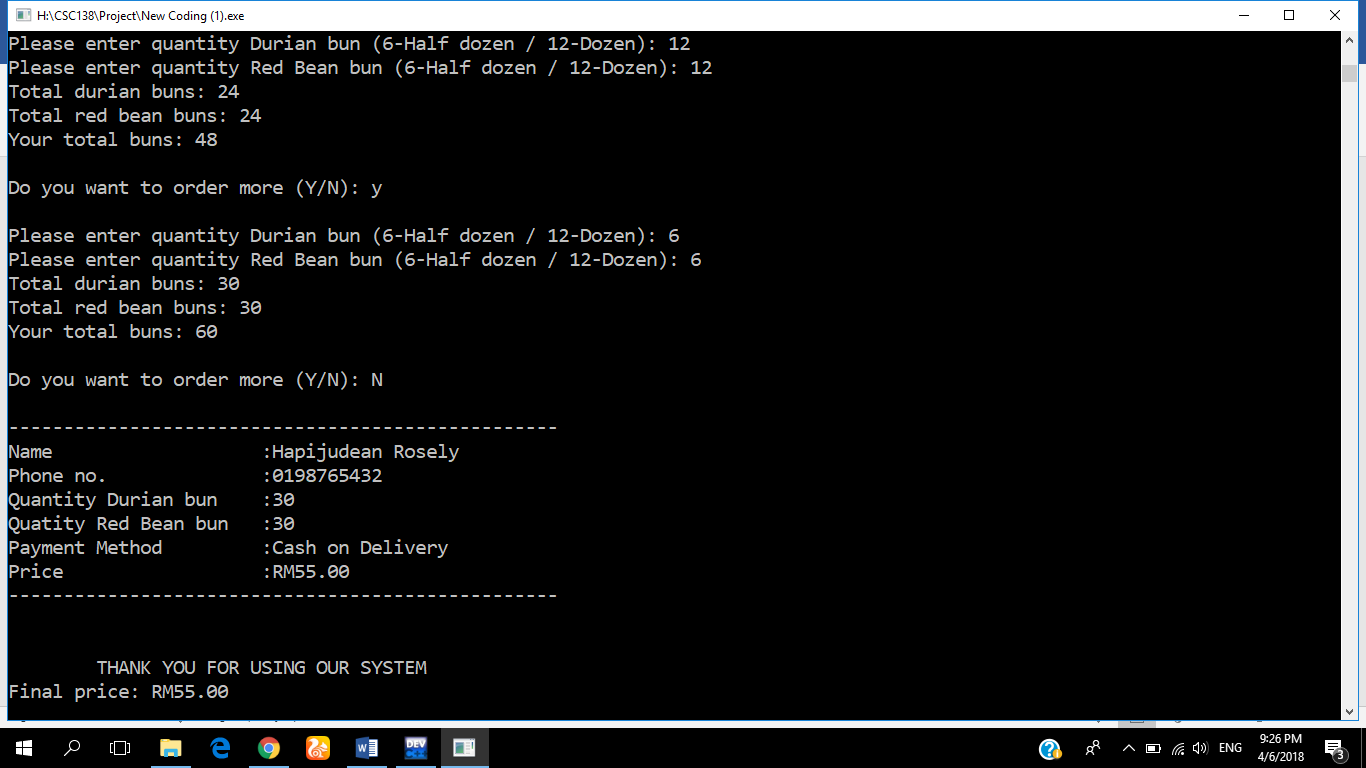


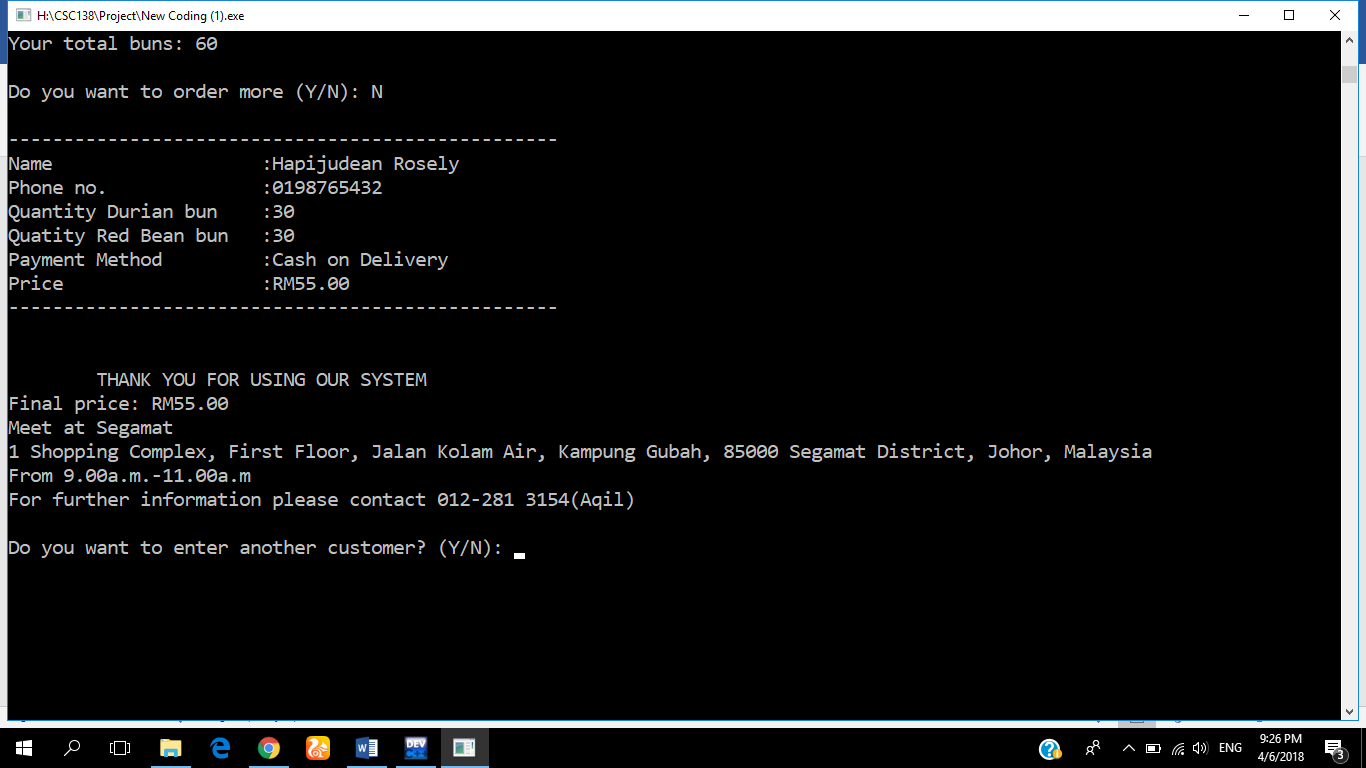


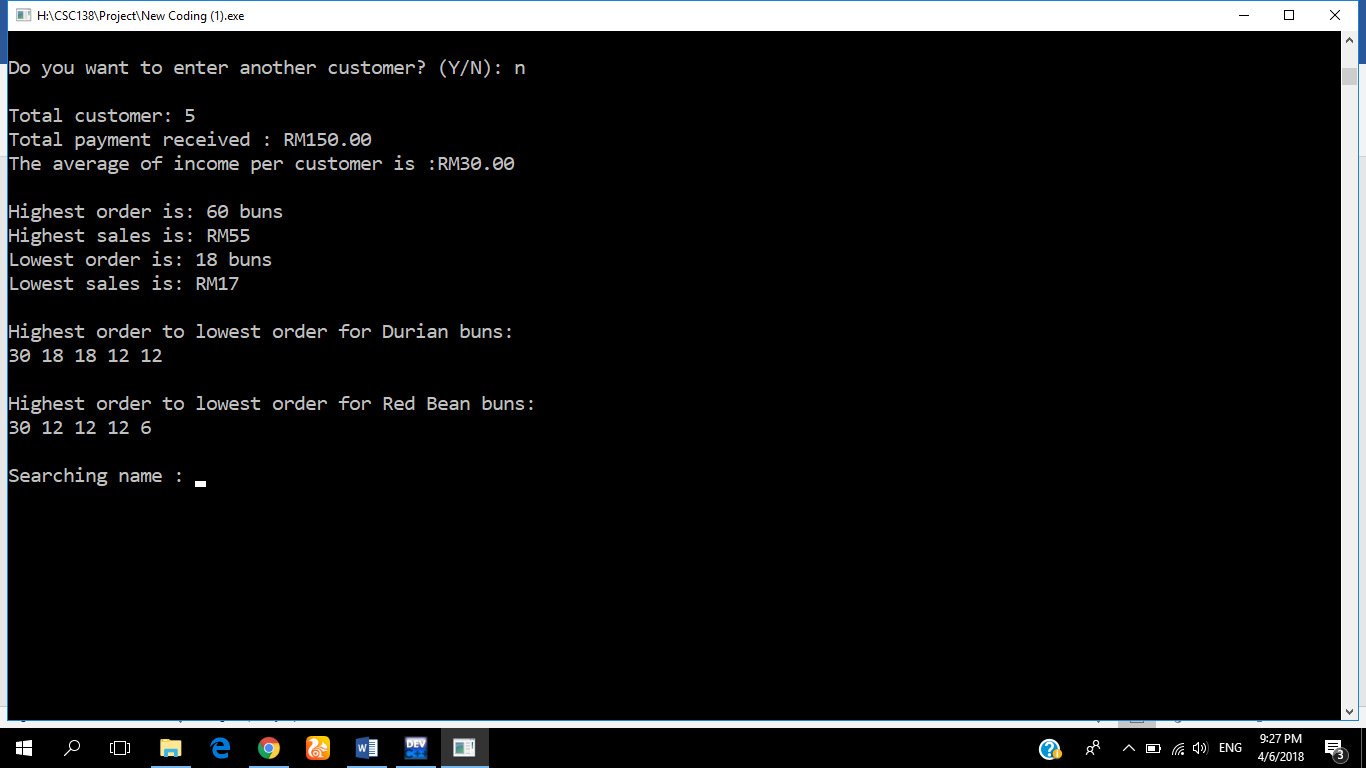


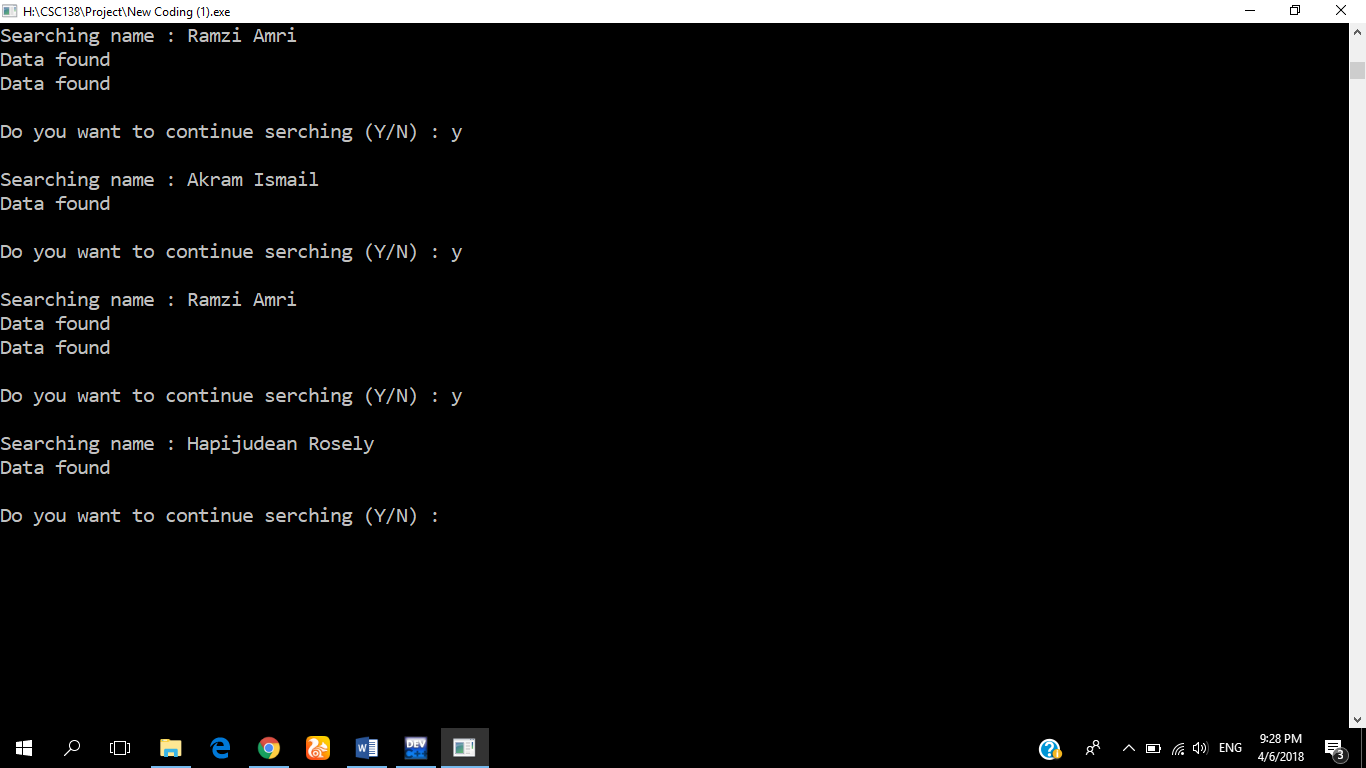


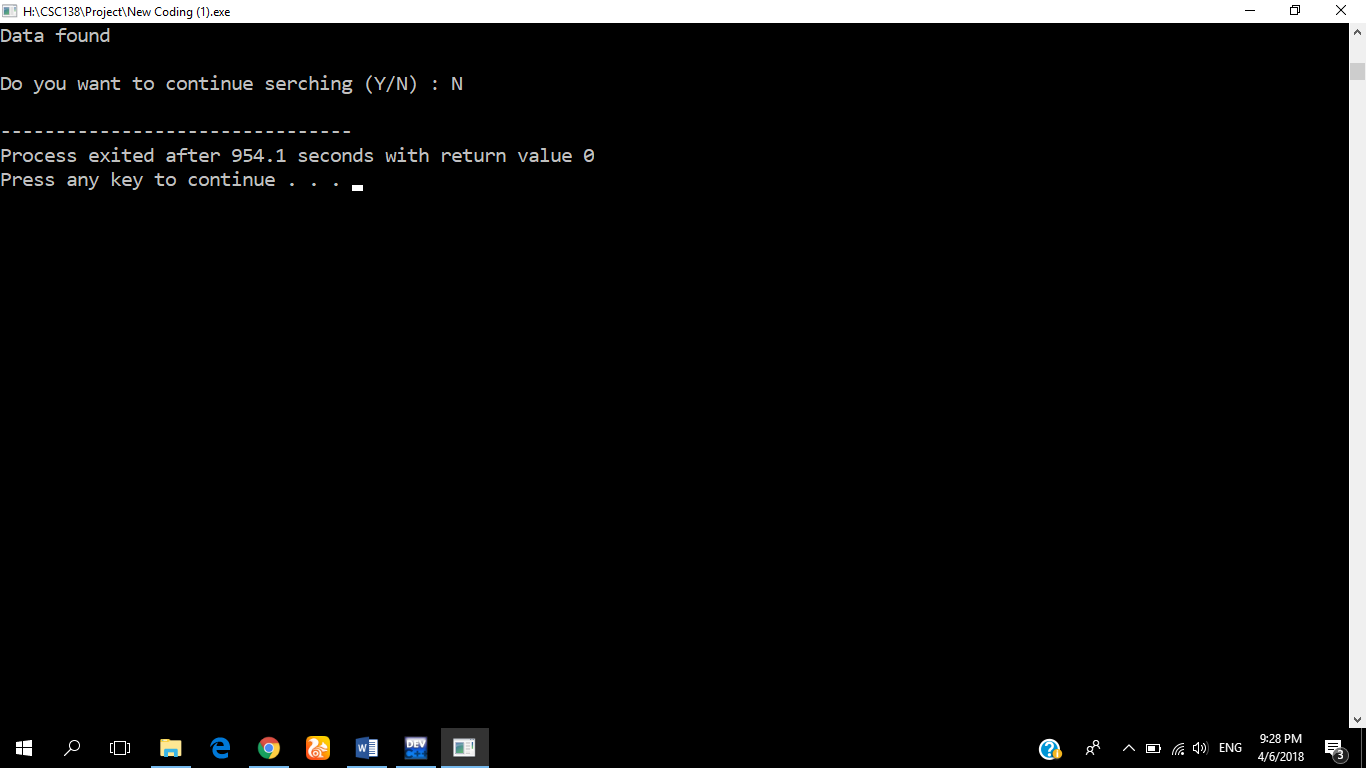




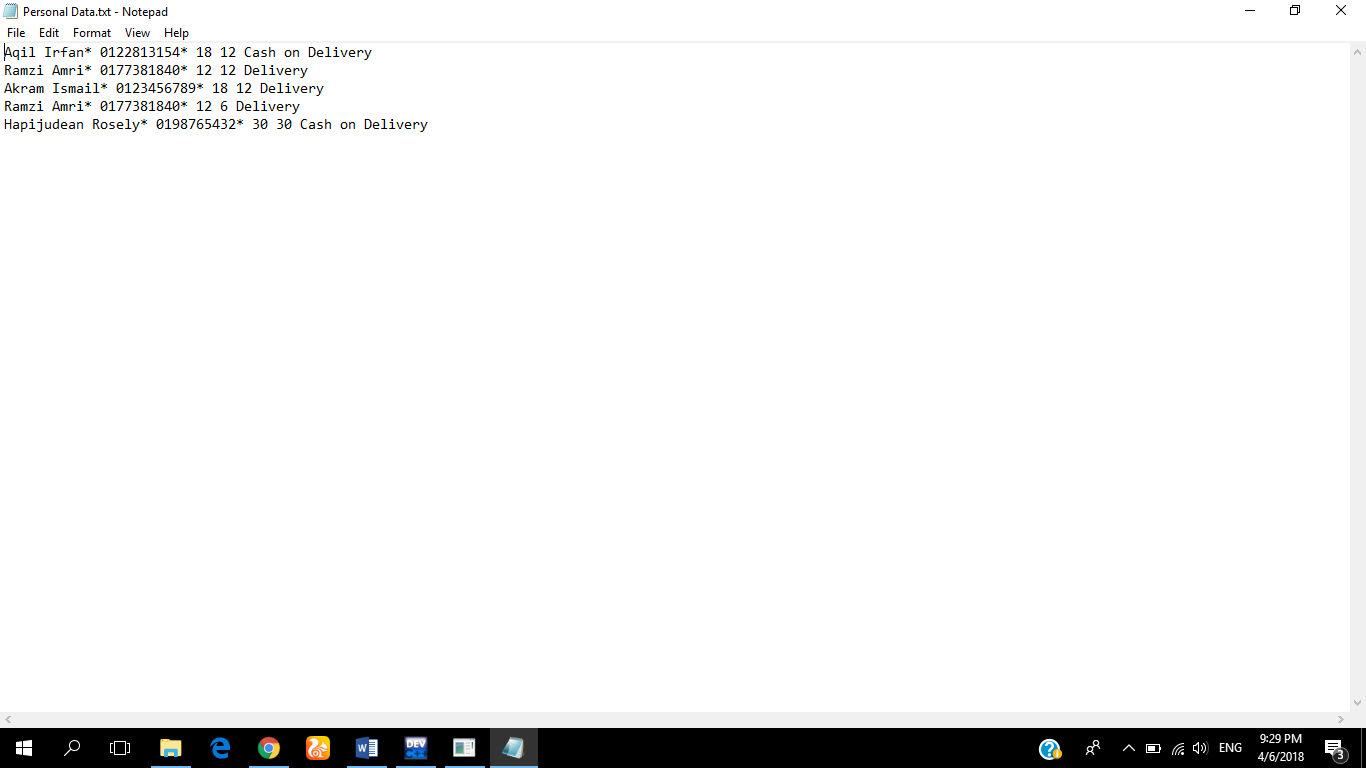




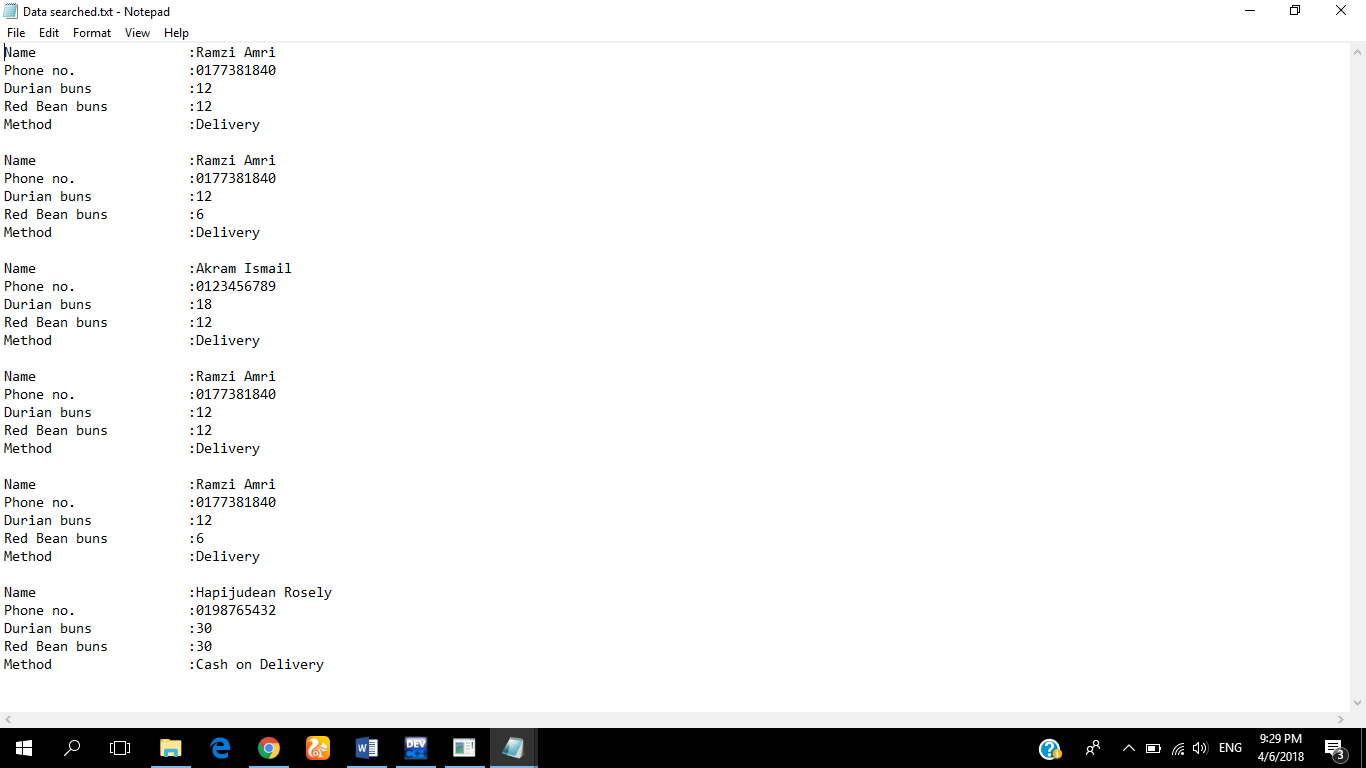




## 4.1 PERSONAL DATA



## 4.2 DATA SEARCHED



## 4.3 LIST CUSTOMER DATA

