****

CSC138 – STRCTURED PROGRMMING PROJECT

**CREATIVE PHONECASE**

LECTURER’S NAME: MADAM RASHIDAH BINTI MOKHTAR

CLASS: JSC1102F

SUBMISSION DATE:

|  |  |
| --- | --- |
| NAME | MATRIX NUMBER |
| NURAIN NABILAH BINTI SALEHUDDIN | 2018258486 |
| ZAFIRA SOFEAH BINTI ZAFRULALLAH | 2018422162 |
| NURAZEEMA BINTI BAHARUDDIN | 2018271134 |

TABLE OF CONTENTS

|  |  |
| --- | --- |
| CONTENTS | PAGES |
| 1.0 Introduction | - |
| 1.1 Project Background | 3 |
| 1.2 Objectives | 4 |
| 1.3 Scope | 5-6 |
| 2.0 Implementation | - |
| 2.1 Usage of Functions | 7 |
| 2.2 Usage of Minimum & Maximum | 8 |
| 2.3 Usage of Counter & Total & Average | 9 |
| 2.4 Usage of Sorting | 10 |
| 2.5 Usage of Searching | 11 |
| 2.6 Usage of Records | 12 |
| 2.7 Usage of Text file | 13 |
| 3.0 Program Testing | - |
| Source Code | 14-29 |
| Sample Input & Output | 30-33 |
| 4.0 Conclusion | 34 |
| 5.0 Refrences | 35 |
| 6.0 CD Pocket | 36 |

1.0 – INTRODUCTION

1.1 Project Background

First and foremost we would like to appreciate and thanked to our beloved Lecturer in charge for our class, Madam Rashidah Binti Mokhtar for guiding and leading us on this whole project for our CSC138 which is Structured Programming. This group consist of three members who are Nurain Nabilah Bt Salehuddin, Zafirah Sofeah Binnti Zafrulallah and Nurazeema Binti Baharuddin. This project proposal report started on 15th March 2019.

The title of our project is “CREATIVE PHONECASE” where we provide designing and printing service for the phone casing. This program lists multiple choices of types of phone model (IPHONE or ANDROID) and types of material (Plastic, Thermo-polyurethane, and Carbon Fiber).There will also be input and output for shipping activities such as; statements of personal information such as name, ID number, IC numbers, address.

This program also include a system that will calculate total price of the item purchased by customers. The price is according to the model of the phone, material of the phone cases. It will also calculate the selling price after discounts if they are a member of the store. This system will also calculate the sum of the whole price including the charges (delivery + the store’s own charges) with a fixed price which is RM10.

At the end of purchasing process the customers will see a receipt of the total price of the items they bought and other information related to their purchases. This program will also display daily report showing the total customer, total sales, average sales, total phone cases sold per day and other related informations.

1.2 Objectives

• To calculate the total price of all items bought by a customer.

• To calculate the total of items sold in a day.

• To calculate the total of each item according to their categories.

• To calculate the total sales for a day.

• To calculate average total sales per day.

• To display customer’s orders (receipt).

• To display daily report per day.

• To find and display minimum and maximum of material of the phone cases.

• To sort and display the phone cases material in a descending order.

• To record all of the customer’s history of buying items from this shop.

• To search and display customer’s information.

1.3 Scope & Program content

SCOPE:

Our targeted users are most millennials generation as this generation love following new trends and always open to new creative world. They are always the ones who are interested in showing off their identity using their phonecase designs it is also one of a way to show their interest in something.

As a qoute taken from a website as a reference “This trend is about life not just fashion.” And the escalating importance of how our phones look reflects the fact that the phone is as much a part of our visual world as our functional one.

For the users of this program, they would find it is more easier and efficient to find design that are suitable to their likings or they can even design according to their own creative ideas.

PROGRAM CONTENT:

|  |  |  |
| --- | --- | --- |
| CONTENT | DESCRIPTION | FURTHER DESCRIPTION |
| Items | Phonecases | - |
| Classification | Types of Phone Model | Iphone or android. |
|  | Types of Material | Plastic,Thermo-polyurathane,Carbon Fiber |
| Calculation | Price | Affordabble for customers |
| Discounts | Membership |
| Charges | Delivery and the store’s own charges |
| Sales | Total sales for a day |
| Counting | * Total customers * Total phonecases sold |

ABLE ON THE PRICING FOR THE ITEMS:

|  |  |  |  |
| --- | --- | --- | --- |
| Phone’s  Model | Case’s  Material | Original Price | Price With Charges (+RM10) |
| *ANDROID*  *or*  *IPHONE* | Plastic | RM20 | RM30 |
| Thermo-polyurathane | RM35 | RM45 |
| Carbon fiber | RM25 | RM35 |

TABLE ON THE CALCULATION:

|  |  |  |  |
| --- | --- | --- | --- |
| Classification | Types of Calculation | Formula | Formula (Coding) |
| Price with charges | -Summation | sellingPrice =  price + RM10 | sellingPrice =  price + 10.00; |
| Total Price  (no discount) | -Summation | totPrice = 0  totPrice = totPrice + sellingPrice | totPrice = 0;  totPrice = totPrice + sellingPrice; |
| Discount | -Multiplication | Discount =  20% x totPrice | Discount =  0.20 \* totPrice; |
| Price with discount | -Subtraction | totalPrice=  totPrice – Discount | totalPrice =  totPrice - Discount |
| Total Sales | -Summation | totSales = 0  totSales = totSales + totalPrice | totSales = 0;  totSales = totSales + totalPrice; |

2.0 – IMPLEMENTATIONS

2.1 – Usage of Functions

* This program uses int main(), functions without return value, functions with return value and functions with reference parameter.
* List of functions:

1. Functions without return value

* void displayHello(ofstream &print)
* void displayOffer(ofstream &print)
* void displayCharges(ofstream &print)
* void displayLoading()
* void displayPrice(ofstream &print)

1. Functions with return value

* int findHighMat(int countPL, int countTP, int countCF)
* int findLowMat(int countPL, int countTP, int countCF)
* double calcSelling(double price)

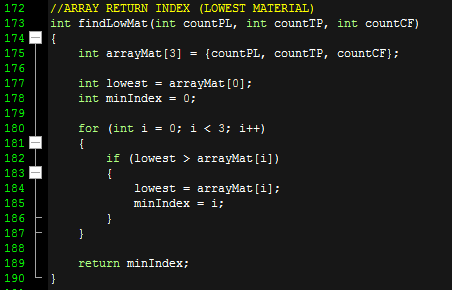
1. Functions with reference parameter

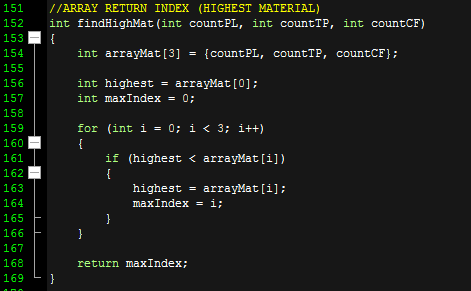
* void calcDiscount(double totPrice, double off,double& discount)
* void sorting (ofstream &print,int& countPL, int& countTP, int& countCF)
* void displayReport(ofstream &print,double totSales, int totCase,double averageSales, int totPL, int totTP, int totCF, int indexMaxMat, int indexMinMat,int& countPL, int& countTP, int& countCF, int size)
* void searching (custInfo customer[],ifstream& find,ofstream& print,int size)
* void readInput (custInfo customer[], ifstream &input,int& totCust)
* void custSelect(custInfo customer[],ifstream &input2,double& price, int& countPL, int& countTP,int& countCF,int count,int& totCust)
* void displayReceipt (custInfo customer[],ofstream &print, int count, char\* disText,int& totCust, int size)

1. int main()

2.2 – Usage of Minimum & Maximum

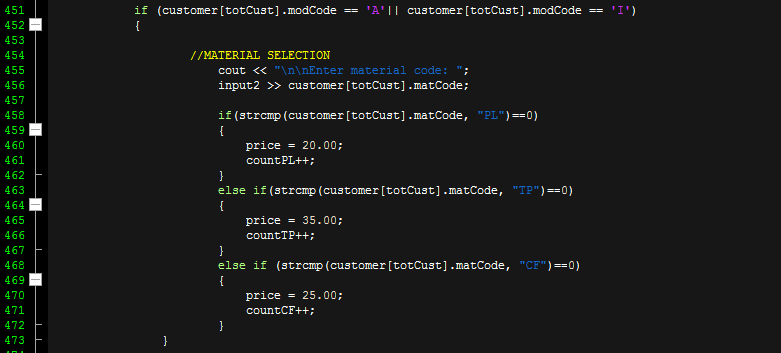
* This program uses array to find the minimum and maximum value of the material of the phone cases such as plastic, thermo-polyurethane and carbon fiber.
* After getting the highest and lowest material sold, it will then display the name of the material at the daily report receipt.



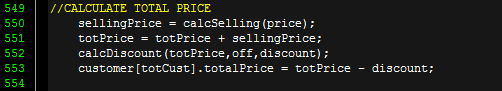


2.3 – Usage of Counter & Total & Average

* This program uses counter to count phone cases based on material per day.



* This program uses total to count the total price per customer and total sales per day.





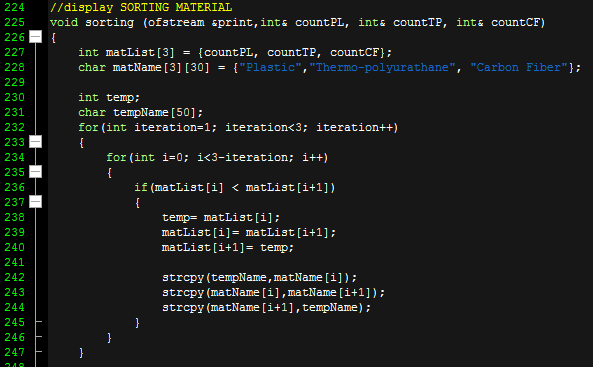
* This program uses average to calculate average sales per day.



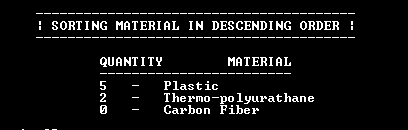
2.4 – Usage of Sorting

* This program uses sorting to sort total of phone cases sold by its’ material such as Plastic, Thermo-polyurethane and Carbon Fiber in a descending order.
* After the system sort the items, at the daily report, the name and its’ total will be displayed.

(\*) sample syntax



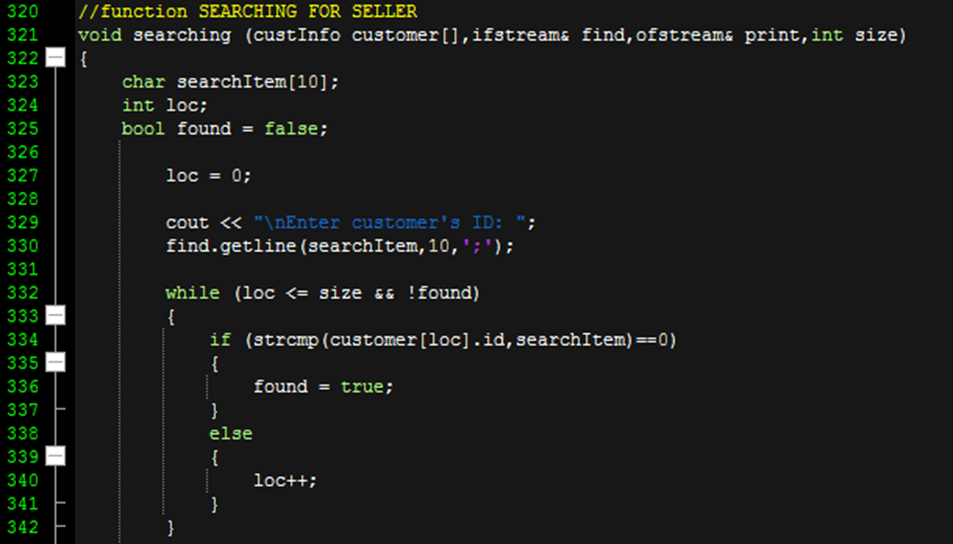
(\*) sample output (daily report)



2.5 – Usage of Searching

* This program uses searching to find customers’ information and purchase history by the customer’s ID code.
* If the ID code exists, the information will be displayed.

(\*) sample syntax



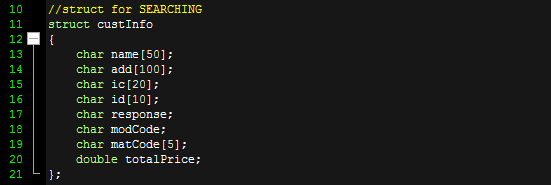
(\*) sample output



2.6 – Usage of Records

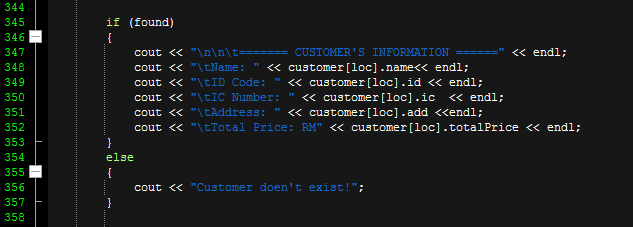
* This program uses struct to make records for each customer’s purchases.
* This struct is used mainly to store various data for searching purposes.

(\*) sample syntax





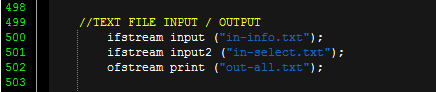
(\*) sample usage of struct



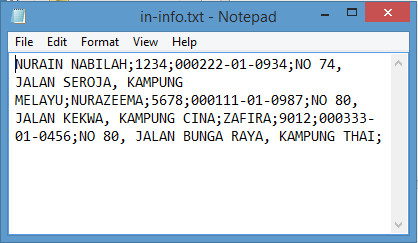
2.7 – Usage of Text file

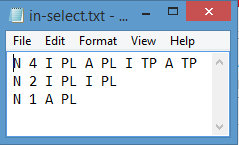
* This program uses text file as input data of customer and output of receipt and daily report.

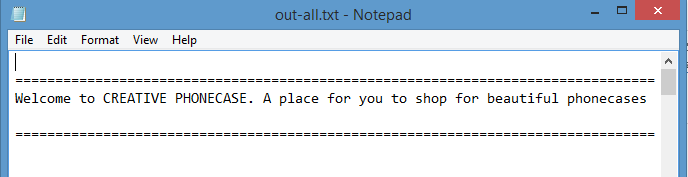
(\*) sample syntax



(\*) sample text file(s)







3.0 – PROGRAM TESTING

3.1 – Source Code

#include <iostream>

#include <iomanip>

#include <cstring>

#include <unistd.h>

#include <Windows.h>

#include <fstream>

**using** **namespace** std**;**

***//struct for SEARCHING***

**struct** custInfo

**{**

**char** name**[**50**];**

**char** add**[**100**];**

**char** ic**[**20**];**

**char** id**[**10**];**

**char** response**;**

**char** modCode**;**

**char** matCode**[**5**];**

**double** totalPrice**;**

**};**

**void** displayHello**(**ofstream **&**print**)**

**{**

cout **<<** **"\n================================================================================";**

cout **<<** **"\nWelcome to CREATIVE PHONECASE. A place for you to shop for beautiful phonecases"** **<<** endl**;**

cout **<<** **"\n================================================================================"** **<<** endl **<<** endl**;**

print **<<** **"\n================================================================================";**

print **<<** **"\nWelcome to CREATIVE PHONECASE. A place for you to shop for beautiful phonecases"** **<<** endl**;**

print **<<** **"\n================================================================================"** **<<** endl **<<** endl**;**

**}**

***//DISCOUNT = 20% for membership***

**void** displayOffer**(**ofstream **&**print**)**

**{**

cout **<<** **"\n\n -------------------- S P E C I A L O F F E R ---------------------- "** **<<** endl**;**

cout **<<** **"\n 20% OFF IS GIVEN TO THE MEMBERS OF CREATIVE PHONECASE "** **<<** endl**;**

cout **<<** **"\n ---------------------------------------------------------------------"** **<<** endl**;**

print **<<** **"\n\n -------------------- S P E C I A L O F F E R ---------------------- "** **<<** endl**;**

print **<<** **"\n 20% OFF IS GIVEN TO THE MEMBERS OF CREATIVE PHONECASE "** **<<** endl**;**

print **<<** **"\n ---------------------------------------------------------------------"** **<<** endl**;**

**}**

***//CHARGES***

**void** displayCharges**(**ofstream **&**print**)**

**{**

cout **<<** **"\n\n\n ---------------------- S H O P C H A R G E S -------------------- "** **<<** endl**;**

cout **<<** **"\n RM 10 WILL BE ADDED TO EACH PHONECASES BOUGHT "** **<<** endl**;**

cout **<<** **"\n -------------------------------------------------------------------"** **<<** endl**;**

print **<<** **"\n\n\n ---------------------- S H O P C H A R G E S -------------------- "** **<<** endl**;**

print **<<** **"\n RM 10 WILL BE ADDED TO EACH PHONECASES BOUGHT "** **<<** endl**;**

print **<<** **"\n -------------------------------------------------------------------"** **<<** endl**;**

**}**

***//LOADING ANIMATIONS***

**void** displayLoading**()**

**{**

system**("COLOR 0E"** **);**

**char** d**=**177**,** c**=**219**;**

cout **<<** **"\n\n\n\t\t\t\tLoading...";**

cout **<<** **"\n\n\n";**

cout **<<** **"\t\t\t";**

**for** **(int** i**=**0**;** i **<=**25**;** i**++)**

cout **<<** d**;**

Sleep**(**150**);**

cout **<<"\r";**

cout **<<"\t\t\t";**

**for** **(int** i**=**0**;** i **<=** 25**;** i**++)**

**{**

cout **<<**c**;**

Sleep**(**200**);**

**}**

cout **<<** **"\n\n\n\n\n";**

system **("COLOR 0F");**

**}**

***//PRICE ANIMATION***

**void** displayPrice**(**ofstream **&**print**)**

**{**

cout **<<** **"\n\n\t .\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. \t .\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. ";**

cout **<<** **"\n\t | | \t | | ";**

cout **<<** **"\n\t | @ ============== | \t | @ ============== | ";**

cout **<<** **"\n\t | I-PHONE (I) | \t | ANDROID (A) | ";**

cout **<<** **"\n\t | .---------------------. | \t | .---------------------. | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | PRICES | | \t | | PRICES | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | MATERIAL: | | \t | | MATERIAL: | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | PLASTIC: | | \t | | PLASTIC: | | ";**

cout **<<** **"\n\t | | RM 20 | | \t | | RM 20 | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | THERMOPOLYURATHANE: | | \t | | THERMOPOLYURATHANE: | | ";**

cout **<<** **"\n\t | | RM 35 | | \t | | RM 35 | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | CARBON FIBER: | | \t | | CARBON FIBER: | | ";**

cout **<<** **"\n\t | | RM 25 | | \t | | RM 25 | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | | | | \t | | | | ";**

cout **<<** **"\n\t | |.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | \t | |.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | ";**

cout **<<** **"\n\t | | \t | | ";**

cout **<<** **"\n\t | |\_\_)---------(\_\_| | \t | |\_\_)---------(\_\_| | ";**

cout **<<** **"\n\t | | \t | | ";**

cout **<<** **"\n\t .-------------------------. \t .-------------------------. ";**

print **<<** **"\n\n\t .\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. \t .\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. ";**

print **<<** **"\n\t | | \t | | ";**

print **<<** **"\n\t | @ ============== | \t | @ ============== | ";**

print **<<** **"\n\t | I-PHONE (I) | \t | ANDROID (A) | ";**

print **<<** **"\n\t | .---------------------. | \t | .---------------------. | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | PRICES | | \t | | PRICES | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | MATERIAL: | | \t | | MATERIAL: | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | PLASTIC: | | \t | | PLASTIC: | | ";**

print **<<** **"\n\t | | RM 20 | | \t | | RM 20 | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | THERMOPOLYURATHANE: | | \t | | THERMOPOLYURATHANE: | | ";**

print **<<** **"\n\t | | RM 35 | | \t | | RM 35 | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | CARBON FIBER: | | \t | | CARBON FIBER: | | ";**

print **<<** **"\n\t | | RM 25 | | \t | | RM 25 | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | | | | \t | | | | ";**

print **<<** **"\n\t | |.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | \t | |.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. | ";**

print **<<** **"\n\t | | \t | | ";**

print **<<** **"\n\t | |\_\_)---------(\_\_| | \t | |\_\_)---------(\_\_| | ";**

print **<<** **"\n\t | | \t | | ";**

print **<<** **"\n\t .-------------------------. \t .-------------------------. ";**

**}**

***//ARRAY RETURN INDEX (HIGHEST MATERIAL)***

**int** findHighMat**(int** countPL**,** **int** countTP**,** **int** countCF**)**

**{**

**int** arrayMat**[**3**]** **=** **{**countPL**,** countTP**,** countCF**};**

**int** highest **=** arrayMat**[**0**];**

**int** maxIndex **=** 0**;**

**for** **(int** i **=** 0**;** i **<** 3**;** i**++)**

**{**

**if** **(**highest **<** arrayMat**[**i**])**

**{**

highest **=** arrayMat**[**i**];**

maxIndex **=** i**;**

**}**

**}**

**return** maxIndex**;**

**}**

***//ARRAY RETURN INDEX (LOWEST MATERIAL)***

**int** findLowMat**(int** countPL**,** **int** countTP**,** **int** countCF**)**

**{**

**int** arrayMat**[**3**]** **=** **{**countPL**,** countTP**,** countCF**};**

**int** lowest **=** arrayMat**[**0**];**

**int** minIndex **=** 0**;**

**for** **(int** i **=** 0**;** i **<** 3**;** i**++)**

**{**

**if** **(**lowest **>** arrayMat**[**i**])**

**{**

lowest **=** arrayMat**[**i**];**

minIndex **=** i**;**

**}**

**}**

**return** minIndex**;**

**}**

***//calc TOTAL CHARGES FOR 1 CUST***

**double** calcSelling**(double** price**)**

**{**

**double** charges**,**sellingPrice**;**

charges **=** 10.00**;**

sellingPrice **=** price **+** charges**;**

**return** sellingPrice**;**

**}**

***//calc DISCOUNT FOR MEMBER***

**void** calcDiscount**(double** totPrice**,** **double** off**,double&** discount**)**

**{**

discount **=** off **\*** totPrice**;**

**}**

***//display TEXT DISCOUNT (20% 0r 0%)***

**void** displayMember**(**custInfo customer**[],** **char\*** disText**,int&** totCust**)**

**{**

**if** **(**customer**[**totCust**].**response **==** 'Y'**)**

**{**

strcpy **(**disText**,** **"20% OFF");**

**}**

**else** **if** **(**customer**[**totCust**].**response **==** 'N'**)**

**{**

strcpy **(**disText**,** **"0% OFF");**

**}**

**}**

***//display SORTING MATERIAL***

**void** sorting **(**ofstream **&**print**,int&** countPL**,** **int&** countTP**,** **int&** countCF**)**

**{**

**int** matList**[**3**]** **=** **{**countPL**,** countTP**,** countCF**};**

**char** matName**[**3**][**30**]** **=** **{"Plastic","Thermo-polyurathane",** **"Carbon Fiber"};**

**int** temp**;**

**char** tempName**[**50**];**

**for(int** iteration**=**1**;** iteration**<**3**;** iteration**++)**

**{**

**for(int** i**=**0**;** i**<**3**-**iteration**;** i**++)**

**{**

**if(**matList**[**i**]** **<** matList**[**i**+**1**])**

**{**

temp**=** matList**[**i**];**

matList**[**i**]=** matList**[**i**+**1**];**

matList**[**i**+**1**]=** temp**;**

strcpy**(**tempName**,**matName**[**i**]);**

strcpy**(**matName**[**i**],**matName**[**i**+**1**]);**

strcpy**(**matName**[**i**+**1**],**tempName**);**

**}**

**}**

**}**

cout **<<** **"\n\n\n\t\t----------------------------------------"** **<<** endl**;**

cout **<<** **"\t\t| SORTING MATERIAL IN DESCENDING ORDER |"** **<<** endl**;**

cout **<<** **"\t\t----------------------------------------"** **<<** endl**;**

cout **<<** **"\n\t\t\tQUANTITY\tMATERIAL";**

cout **<<** **"\n\t\t\t------------------------"** **<<** endl**;**

**for** **(int** i **=** 0**;** i **<** 3**;** i**++)**

**{**

cout **<<"\t\t\t"** **<<** matList**[**i**]** **<<** **" - "** **<<** matName**[**i**]** **<<** endl**;**

**}**

print **<<** **"\n\n\n\t\t----------------------------------------"** **<<** endl**;**

print **<<** **"\t\t| SORTING MATERIAL IN DESCENDING ORDER |"** **<<** endl**;**

print **<<** **"\t\t----------------------------------------"** **<<** endl**;**

print **<<** **"\n\t\t\tQUANTITY\tMATERIAL";**

print **<<** **"\n\t\t\t------------------------"** **<<** endl**;**

**for** **(int** i **=** 0**;** i **<** 3**;** i**++)**

**{**

print **<<"\t\t\t"** **<<** matList**[**i**]** **<<** **" - "** **<<** matName**[**i**]** **<<** endl**;**

**}**

**}**

***//display DAILY REPORT***

**void** displayReport**(**ofstream **&**print**,double** totSales**,** **int** totCase**,double** averageSales**,** **int** totPL**,** **int** totTP**,** **int** totCF**,** **int** indexMaxMat**,** **int** indexMinMat**,int&** countPL**,** **int&** countTP**,** **int&** countCF**,const** **int** size**)**

**{**

**char** material**[**3**][**20**]** **=** **{** **"Plastic",** **"Thermo-polyurathane",** **"Carbon Fiber"};**

cout **<<** **"\n\n\t\tWelcome ADMIN! You have worked hard today.";**

cout **<<** **"\n\t\tThis the report of the day!";**

cout **<<** setprecision **<<** fixed**;**

cout **<<** **"\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

cout **<<** **"\t - DAILY REPORT ON SALES - "** **<<** endl**;**

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

cout **<<** **"\n\tTotal customers : "** **<<** size**;**

cout **<<** **"\n\n\tTotal Sales : RM "** **<<** totSales**;**

cout **<<** **"\n\n\tAverage Sales : RM "** **<<** averageSales**;**

cout **<<** **"\n\n\tTotal phonecases sold : "** **<<** totCase**;**

cout **<<** **"\n\n\n\tTotal Plastic Material : "** **<<** totPL**;**

cout **<<** **"\n\n\tTotal Thermo-polyurathane Material : "** **<<** totTP**;**

cout **<<** **"\n\n\tTotal Carbon Fiber Material : "** **<<** totCF**;**

cout **<<** **"\n\n\n\tHighest Material : "** **<<** material**[**indexMaxMat**];**

cout **<<** **"\n\n\tLowest Material : "** **<<** material**[**indexMinMat**];**

cout **<<** **"\n\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

cout **<<** **"\t See you again! "** **<<** endl**;**

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

print **<<** **"\n\n\t\tWelcome ADMIN! You have worked hard today.";**

print **<<** **"\n\t\tThis the report of the day!";**

cout **<<** setprecision **<<** fixed**;**

print **<<** **"\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\t - DAILY REPORT ON SALES - "** **<<** endl**;**

print **<<** **"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\n\tTotal customers : "** **<<** size**;**

print **<<** **"\n\n\tTotal Sales : RM "** **<<** totSales**;**

print **<<** **"\n\n\tAverage Sales : RM "** **<<** averageSales**;**

print **<<** **"\n\n\tTotal phonecases sold : "** **<<** totCase**;**

print **<<** **"\n\n\n\tTotal Plastic Material : "** **<<** totPL**;**

print **<<** **"\n\n\tTotal Thermo-polyurathane Material : "** **<<** totTP**;**

print **<<** **"\n\n\tTotal Carbon Fiber Material : "** **<<** totCF**;**

print **<<** **"\n\n\n\tHighest Material : "** **<<** material**[**indexMaxMat**];**

print **<<** **"\n\n\tLowest Material : "** **<<** material**[**indexMinMat**];**

sorting **(**print **,**countPL**,** countTP**,** countCF**);**

print **<<** **"\n\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\t See you again! "** **<<** endl**;**

print **<<** **"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl **<<** endl**;**

**}**

***//function SEARCHING FOR SELLER***

**void** searching **(**custInfo customer**[],**ifstream**&** find**,**ofstream**&** print**,int** size**)**

**{**

**char** searchItem**[**10**];**

**int** loc**;**

**bool** found **=** **false;**

loc **=** 0**;**

cout **<<** **"\nEnter customer's ID: ";**

find**.**getline**(**searchItem**,**10**,**';'**);**

**while** **(**loc **<=** size **&&** **!**found**)**

**{**

**if** **(**strcmp**(**customer**[**loc**].**id**,**searchItem**)==**0**)**

**{**

found **=** **true;**

**}**

**else**

**{**

loc**++;**

**}**

**}**

**if** **(**found**)**

**{**

cout **<<** **"\n\n\t======= CUSTOMER'S INFORMATION ======"** **<<** endl**;**

cout **<<** **"\tName: "** **<<** customer**[**loc**].**name**<<** endl**;**

cout **<<** **"\tID Code: "** **<<** customer**[**loc**].**id **<<** endl**;**

cout **<<** **"\tIC Number: "** **<<** customer**[**loc**].**ic **<<** endl**;**

cout **<<** **"\tAddress: "** **<<** customer**[**loc**].**add **<<**endl**;**

cout **<<** **"\tTotal Price: RM"** **<<** customer**[**loc**].**totalPrice **<<** endl**;**

**}**

**else**

**{**

cout **<<** **"Customer doen't exist!";**

**}**

**if** **(**found**)**

**{**

print **<<** **"\n\n\t======= CUSTOMER'S INFORMATION ======"** **<<** endl**;**

print **<<** **"\tName: "** **<<** customer**[**loc**].**name**<<** endl**;**

print **<<** **"\tID Code: "** **<<** customer**[**loc**].**id**<<** endl**;**

print **<<** **"\tIC Number: "** **<<** customer**[**loc**].**ic **<<** endl**;**

print **<<** **"\tAddress: "** **<<** customer**[**loc**].**add **<<**endl**;**

print **<<** **"\tTotal Price: RM"** **<<** customer**[**loc**].**totalPrice **<<** endl**;**

**}**

**else**

**{**

print **<<** **"Customer doen't exist!";**

**}**

**}**

***//function INPUT***

**void** readInput **(**custInfo customer**[],** ifstream **&**input**,int&** totCust**)**

**{**

cout **<<** **"\n\nEnter name: ";**

input**.**getline**(**customer**[**totCust**].**name**,**50**,** ';'**);**

cout **<<** **"Enter user ID: ";**

input**.**getline**(**customer**[**totCust**].**id**,**10**,** ';'**);**

cout **<<** **"Enter IC Number: ";**

input**.**getline**(**customer**[**totCust**].**ic**,**20**,** ';'**);**

cout **<<** **"Enter your address: ";**

input**.**getline**(**customer**[**totCust**].**add**,**100**,** ';'**);**

**}**

***//display RECEIPT PER CUST***

**void** displayReceipt **(**custInfo customer**[],**ofstream **&**print**,** **int** count**,** **char\*** disText**,int&** totCust**,** **int** size**)**

**{**

***//output (Receipt)***

cout **<<** setprecision**(**2**)** **<<** fixed**;**

cout **<<** **"\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

cout **<<** **"\t - CREATIVE PHONECASE OFFICIAL RECEIPT - "** **<<** endl**;**

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

cout **<<** **"\n\t BUYER'S PERSONAL INFORMATIONS "** **<<** endl**;**

cout **<<** **"\n\tName : "** **<<** customer**[**totCust**].**name**;**

cout **<<** **"\n\n\tUser ID : "** **<<** customer**[**totCust**].**id**;**

cout **<<** **"\n\n\tTotal phonecases bought : "** **<<** count**;**

cout **<<** **"\n\n\tIC Number : "** **<<** customer**[**totCust**].**ic**;**

cout **<<** **"\n\n\tThe items will delivered to : "** **<<** customer**[**totCust**].**add**;**

cout **<<** **"\n\n\n\t CHARGES AND ITEMS INFORMATIONS "** **<<** endl**;**

cout **<<** **"\n\n\tTax : FREE FROM TAX"** **;**

cout **<<** **"\n\n\tDiscount : "** **<<** disText**;**

cout **<<** **"\n\n\tCharges : RM 10.00 each";**

cout **<<** **"\n\n\tTotal Price : RM "** **<<** customer**[**totCust**].**totalPrice**;**

cout **<<** **"\n\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

cout **<<** **"\t THANK YOU FOR BUYING WITH US! "** **<<** endl**;**

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

cout **<<** setprecision**(**2**)** **<<** fixed**;**

print **<<** **"\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\t - CREATIVE PHONECASE OFFICIAL RECEIPT - "** **<<** endl**;**

print **<<** **"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\n\t BUYER'S PERSONAL INFORMATIONS "** **<<** endl**;**

print **<<** **"\n\tName : "** **<<** customer**[**totCust**].**name**;**

print **<<** **"\n\n\tUser ID : "** **<<** customer**[**totCust**].**id**;**

print **<<** **"\n\n\tTotal phonecases bought : "** **<<** count**;**

print **<<** **"\n\n\tIC Number : "** **<<** customer**[**totCust**].**ic**;**

print **<<** **"\n\n\tThe items will delivered to : "** **<<** customer**[**totCust**].**add**;**

print **<<** **"\n\n\n\t CHARGES AND ITEMS INFORMATIONS "** **<<** endl**;**

print **<<** **"\n\n\tTax : FREE FROM TAX"** **;**

print **<<** **"\n\n\tDiscount : "** **<<** disText**;**

print **<<** **"\n\n\tCharges : RM 10.00 each";**

print **<<** **"\n\n\tTotal Price : RM "** **<<** customer**[**totCust**].**totalPrice**;**

print **<<** **"\n\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

print **<<** **"\t THANK YOU FOR BUYING WITH US! "** **<<** endl**;**

print **<<** **"\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"** **<<** endl**;**

**}**

***//fuction SELECTION***

**void** custSelect**(**custInfo customer**[],**ifstream **&**input2**,double&** price**,** **int&** countPL**,** **int&** countTP**,int&** countCF**,int** count**,int&** totCust**)**

**{**

***//MODEL SELECTION***

cin**.**clear**();**

cout **<<** **"\n\nEnter model code: ";**

input2 **>>** customer**[**totCust**].**modCode**;**

**if** **(**customer**[**totCust**].**modCode **==** 'A'**||** customer**[**totCust**].**modCode **==** 'I'**)**

**{**

***//MATERIAL SELECTION***

cout **<<** **"\n\nEnter material code: ";**

input2 **>>** customer**[**totCust**].**matCode**;**

**if(**strcmp**(**customer**[**totCust**].**matCode**,** **"PL")==**0**)**

**{**

price **=** 20.00**;**

countPL**++;**

**}**

**else** **if(**strcmp**(**customer**[**totCust**].**matCode**,** **"TP")==**0**)**

**{**

price **=** 35.00**;**

countTP**++;**

**}**

**else** **if** **(**strcmp**(**customer**[**totCust**].**matCode**,** **"CF")==**0**)**

**{**

price **=** 25.00**;**

countCF**++;**

**}**

**}**

**}**

**int** main **()**

**{**

***//NORMAL Declaration***

**int** count **=** 0**,**totCase **=** 0**,** limit**;**

**int** countPL **=** 0**,** countTP **=** 0**,** countCF **=** 0**,** totPL**,** totTP**,** totCF**;**

**double** price**,** charges**,**off**,** discount**,** sellingPrice**,** discountPrice**,** averageSales**;**

**double** totSales **=** 0**,** totPrice **=** 0**;**

**char** modName**[**20**],**maxMat**[**20**],** minM**[**20**],** disText**[**10**];**

**const** **int** size **=**3**;**

***//ARRAY Declaration***

**char** tryModel**[**100**][**50**];**

**int** b**=**0**;**

**int** indexMaxMat**,** indexMinMat**;**

**int** matList**[**3**];**

***//STRUCT Declaration***

custInfo customer**[**size**];**

***//TEXT FILE INPUT / OUTPUT***

ifstream input **("in-info.txt");**

ifstream input2 **("in-select.txt");**

ofstream print **("out-all.txt");**

***//WELCOME & PRICE TABLE & DISCOUNT & CHARGES***

displayHello**(**print**);**

displayOffer**(**print**);**

displayPrice**(**print**);**

displayCharges**(**print**);**

***//READ INPUT***

**for** **(int** totCust **=**0**;** totCust **<** size**;** totCust**++)**

**{**

readInput **(**customer**,**input**,**totCust**);**

**}**

**for** **(int** totCust **=**0**;** totCust **<** size**;** totCust**++)**

**{**

***//DISCOUNT OFFER***

cout **<<** **"\n\nDo you have a membership here? < Y - YES / N - NO > ";**

input2 **>>** customer**[**totCust**].**response**;**

cout **<<** **"\n\nHow many do you want to buy?";**

input2 **>>** limit**;**

displayMember**(**customer**,**disText**,**totCust**);**

**for(**count **=** 0**;** count **<** limit**;** count**++)** ***//inner loop for one customer***

**{**

***//SELECTION PROCESS***

custSelect**(**customer**,**input2**,**price**,**countPL**,** countTP**,**countCF**,**count**,**totCust**);**

***//DISCOUNT***

**if** **(**customer**[**totCust**].**response **==** 'Y'**)**

**{**

off **=** 0.20**;**

**}**

**else** **if** **(**customer**[**totCust**].**response **==** 'N'**)**

**{**

off **=** 0.00**;**

**}**

***//CALCULATE TOTAL PRICE***

sellingPrice **=** calcSelling**(**price**);**

totPrice **=** totPrice **+** sellingPrice**;**

calcDiscount**(**totPrice**,**off**,**discount**);**

customer**[**totCust**].**totalPrice **=** totPrice **-** discount**;**

**}** ***//LOOPING ITEM END***

***//RECEIPT***

displayReceipt **(**customer**,**print**,**count**,**disText**,**totCust**,**size**);**

***//CALCULATE DAILY REPORT***

totSales **=** totSales **+** customer**[**totCust**].**totalPrice**;**

totCase **=** totCase **+** count**;**

averageSales **=** totSales**/**size**;**

***//initialize value for count material***

totPL **=** countPL**;**

totTP **=** countTP**;**

totCF **=** countCF**;**

input**.**close**();**

***//reset to zero for next customer***

totPrice **=** 0**;**

***//func call ARRAY (MIN MAX MATERIAL)***

indexMaxMat **=** findHighMat**(**countPL**,** countTP**,** countCF**);**

indexMinMat **=** findLowMat**(**countPL**,** countTP**,** countCF**);**

**}** ***//ONEDAY LOOP ENDS***

***//DAILY REPORT***

displayLoading**();**

displayReport**(**print**,** totSales**,** totCase**,**averageSales **,**totPL**,** totTP**,** totCF**,** indexMaxMat**,** indexMinMat**,**countPL**,**countTP**,**countCF**,**size**);**

***//SEARCHING OPTION***

ifstream find **("in-search.txt");**

**const** **int** cust **=** 2**;**

**for** **(int** i **=**0**;** i **<** cust**;**i**++)**

searching**(**customer**,**find**,**print**,**size**);**

find**.**close**();**

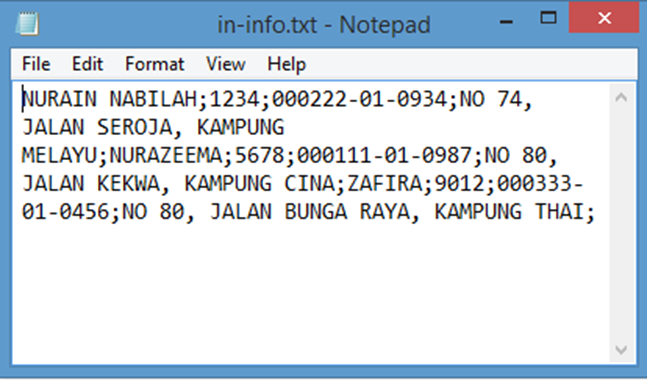
print**.**close**();**

**return** 0**;**

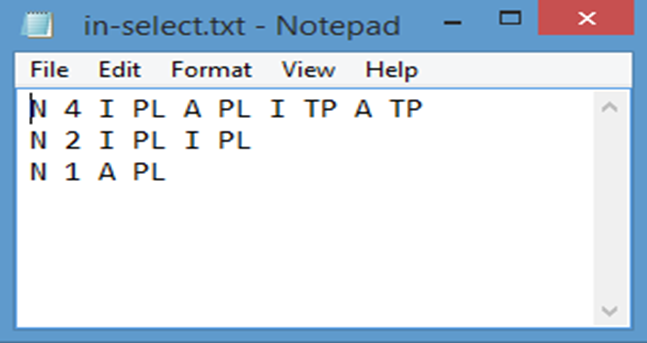
**}**

3.2 – Sample Input & Output

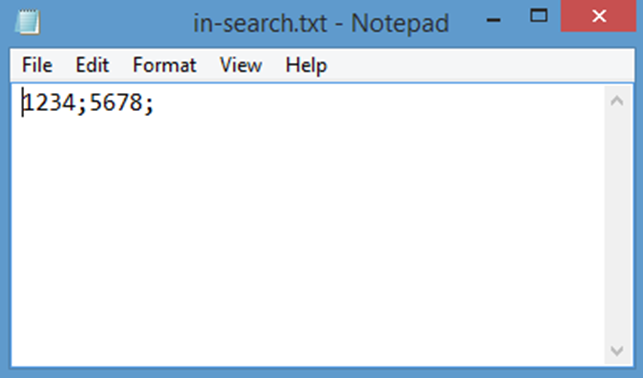
* SAMPLE INPUT(S)
* Input #1 – Customer’s information



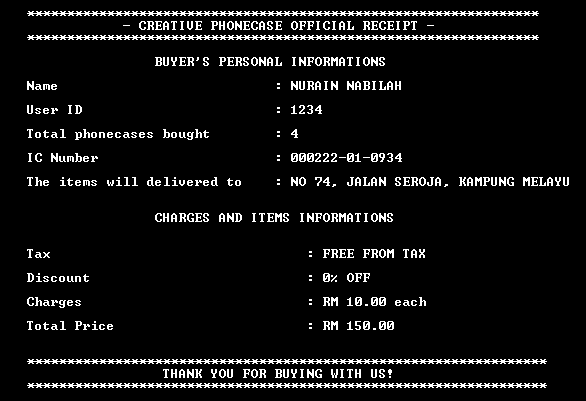
* Input #2 – Customer’s selection

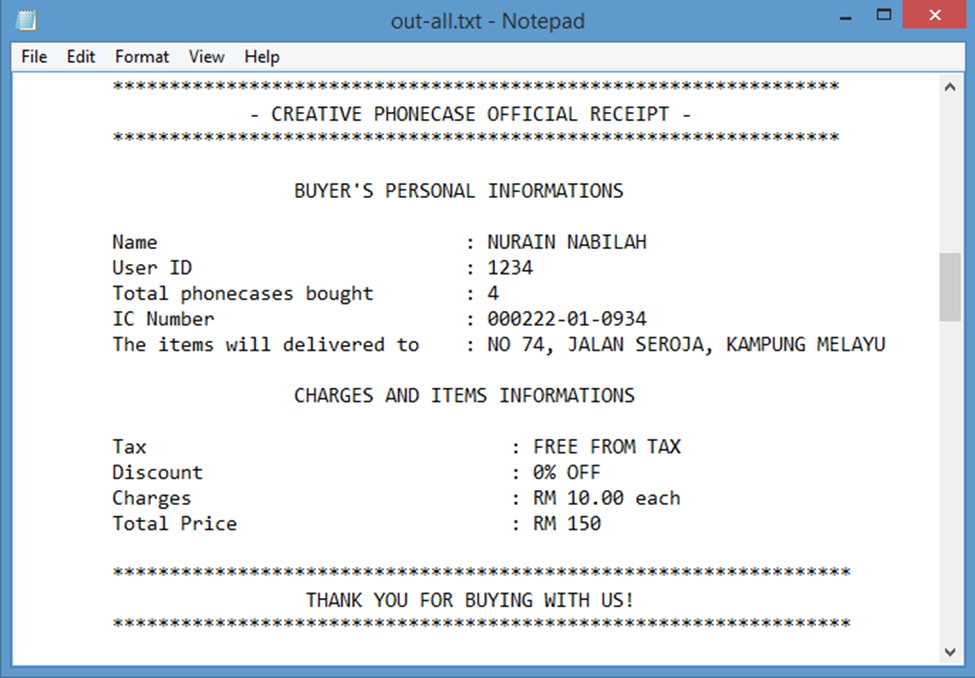


* Input #3 – Employee’s search for customer’s purchases history

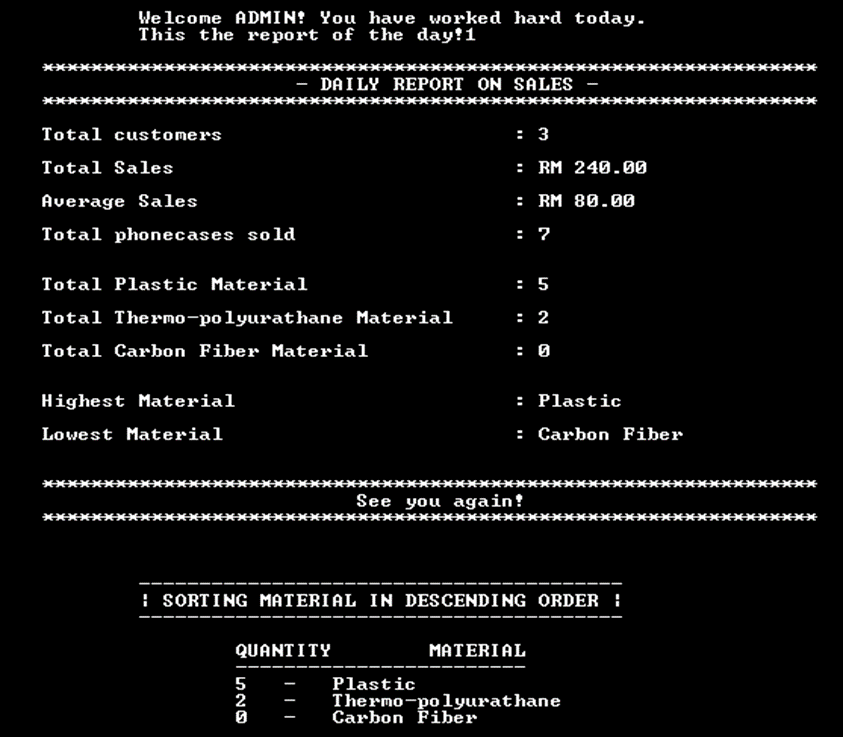


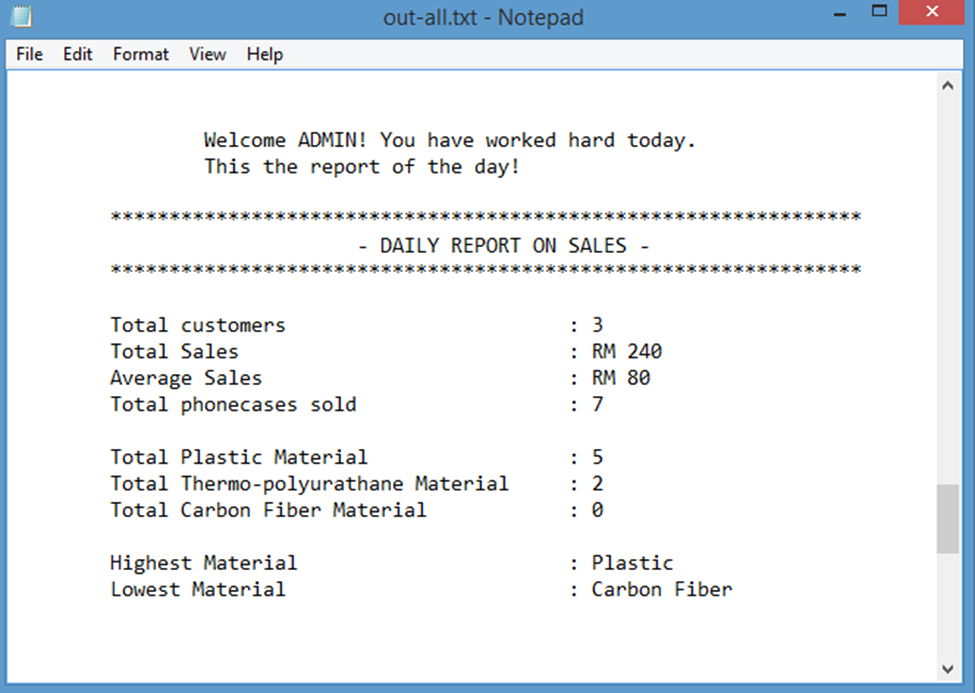
* SAMPLE OUTPUT(S)
* Output #1 – Each customer’s receipt(s)



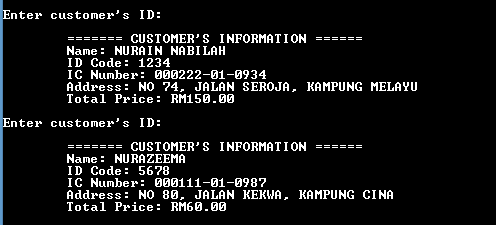


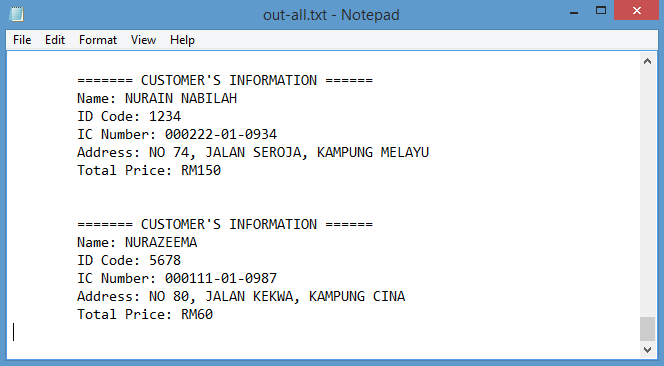
* Output #2 – Daily Report





* Output #3 – Customer’s purchases history





4.0 – CONCLUSION

5.0 – REFERENCES

6.0 – CD POCKET