

N. K. BAGRODIA PUBLIC SCHOOL

[AHINSA MARG, SECTOR-9, ROHINI, DELHI-85]

A Project Report

On

SNACKS PARLOUR

As a part of the Computer Science Course (083)

SUBMITTED BY:

PAARTH SETHI

Class XII

2016-2017

Under the Guidance of:

Ms. Annu Saini

(PGT, Computer Science)

*CERTIFICATE*

This is to certify that the Project entitled, “SNACKS PARLOUR” is a bona fide work done by Master PAARTH SETHI of class XII-A Session 2016-17 has been carried out under my direct supervision and guidance.

This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.



Signature of Teacher/Guide

Name: Ms. ANNU SAINI

Designation: PGT, COMPUTER SCIENCE



*ACKNOWLEDGMENTS*

The pleasant task of expressing gratitude to those who contributed to the completion and presentation of this project is severely constrained by limited space for describing the nature and extent of debts to them.

I take this opportunity to express my sincere thanks and gratitude to Ms. Annu Saini for her efficient guidance, generous attention and encouragement throughout.

I would also like to thank all the teaching and non-teaching staff of Computer Science department who helped me directly or indirectly in the completion of this project.

Last but not the least with affection and regards I express my gratitude and to my parents and loving brother for giving me constant encouragement and support in making this project a success.

Above all, I thank the Almighty Lord for His blessings.

PAARTH SETHI

# Contents

[Introduction 1](#_Toc467791338)

[Objectives 2](#_Toc467791339)

[System Requirements 3](#_Toc467791340)

[List Of All Header File 4](#_Toc467791341)

[List of Classes/Structurer 5](#_Toc467791342)

[List of Functions 6](#_Toc467791343)

[Flow Chart 11](#_Toc467791344)

[Source Code 12](#_Toc467791345)

[Outputs 49](#_Toc467791346)

[Limitations 59](#_Toc467791347)

[Bibliography 60](#_Toc467791348)

# Introduction

**C++ was developed by Bjarne Stroustrupof AT&T Bell Laboratories in the early 1980's, and is based on the C language. The "++" is a syntactic construct used in C (to increment a variable), and C++ is intended as an incremental improvement of C. Most of C is a subset of C++, so that most C programs can be compiled (i.e. converted into a series of low-level instructions that the computer can execute directly) using a C++ compiler.**

C is in many ways hard to categorise. Compared to *assembly* language it is high-level, but it nevertheless includes many low-level facilities to directly manipulate the computer's memory. It is therefore an excellent language for writing efficient "systems" programs. But for other types of programs, C code can be hard to understand, and C programs can therefore be particularly prone to certain types of error. The extra object-oriented facilities in C++ are partly included to overcome these shortcomings.

# Objectives

The project is designed for customer billing system in C++. The title of the project is “Snacks Parlour“. In this project a user can take an order, generate a bill, access details about employees and items. The bill generation will take place at the time of delivery, if the delivery is late that is the order doesn’t reach in a few minutes, 15% of the total amount will be deducted.

Administrator of the project can enter, modify and delete employee, item record, display all/specific employee, item and bill record. Administrator can enter a new user which can access only display, search function and take order or generate bill.

# System Requirements

**HARDWARE SPECIFICATION**

* **MEMORY : 984 KB**
* **MICROPROCESSOR:1.2**
* **HARD DISK:40 GB**
* **PRINTER: HP LASER PRINTER**

**SOFTWARE SPECIFICATION**

* PLATFORM : C++ with or without graphics
* FRONT END: C++
* WINDOWS 7 or above
* MS WORD

# List Of All Header File

|  |  |  |
| --- | --- | --- |
| 1. | fstream.h | This header file contains definitions of functions for file handling, cin and cout |
| 2. | time.h | This header file contains definitions of functions to get and manipulate date and time information. |
| 3. | string.h | This header file contains definitions of functions for string handling |
| 4. | conio.h | This header file contains definitions of functions for clrscr() and getch() functions |
| 5. | stdio.h | This header file contains definitions of functions for standard I/O operations |
| 6. | dos.h | This header file contains definitions of functions for gotoxy() |

# List of Classes/Structure

|  |  |  |
| --- | --- | --- |
| 1. | struct mydate | Structure for accessing Date |
| 2. | class employee | Class for accessing employee details |
| 3. | class itemss | Class for accessing item details |
| 4. | class user | Class for accessing user details |
| 5. | class customer | Class for accessing customer details |
| 6. | class order | Class for accessing order details |
| 7. | class bill | Class for accessing bill details |

# List of Functions

|  |  |  |
| --- | --- | --- |
| 1. | void employee::getdata\_emp(int e) | Function for getting employee data |
| 2. | void employee::showdata\_emp(int j) | Function for showing employee data |
| 3. | int employee::get\_empid() | Function for returning employee ID |
| 4. | void employee::getnewdata\_emp() | Function for getting new employee data |
| 5. | void itemss::getdata\_item(int e) | Function for getting item data |
| 6. | void itemss::showdata\_item(int i) | Function for showing item data |
| 7. | void itemss::showdata\_item1(int r) | Function for showing item data |
| 8. | int itemss::get\_itemid() | Function for returning item ID |
| 9. | float itemss::get\_price() | Function for returning item price |
| 10. | char \*itemss::get\_name() | Function for returning item name |
| 11. | void itemss::getnewdata\_item() | Function for getting new item data |
| 12. | void user::getdata\_user() | Function for getting user data |
| 13. | void user::showdata\_user(int i) | Function for showing user data |
| 14. | char \*user::get\_id() | Function for returning user ID |
| 15. | char \*user::get\_ps() | Function for returning user password |
| 16. | void customer::getdata\_cust(int a,char b[],char c[],char d[],int e) | Function for getting customer data |
| 17. | void customer::showdata\_cust(int i) | Function for showing customer data |
| 18. | int customer::get\_custid() | Function for returning customer ID |
| 19. | char \*customer::get\_name() | Function for returning customer name |
| 20. | int customer::get\_oid() | Function for returning order ID |
| 21. | int \*order::getitemid() | Function for returning item ID |
| 22. | int \*order::getqty() | Function for returning Quantity |
| 23. | float \*order::getamt() | Function for returning total order amount |
| 24. | int order::getsize() | Function for returning total order size |
| 25. | float order::gettotal() | Function for returning total amount |
| 26. | void order::getdata\_order(int a,mydate b,int c[],int d[],float e[],float f,int n) | Function for getting order data |
| 27. | void order::showdata\_order(int j) | Function for showing order data |
| 28. | int order::get\_orderid() | Function for order ID |
| 29. | void bill::getdata\_bill(int a,mydate b,char c,float d) | Function for getting bill data |
| 30. | void bill::showdata\_bill(int i) | Function for showing bill data |
| 31. | int bill::get\_billno() | Function for returning bill data |
| 32. | void box(int r1,int c1,int r2,int c2) | Function for creating A Box |
| 33. | void modifyrecord\_emp() | Function To modify Employee information |
| 34. | void deleterecord\_emp() | Function To delete Employee information |
| 35. | void addrecord\_emp() | Function for adding Employee record |
| 36. | void showallrecord\_emp() | Function to show Employee record |
| 37. | void searchrecord\_emp() | Function to search Employee information |
| 38. | int getlastitemno() | Function To get last Item Number |
| 39. | void addrecord\_item() | Function To add item record |
| 40. | void showallrecord\_item() | Function To display item record |
| 41. | void showallrecord\_item1() | Function To display item record |
| 42. | void searchrecord\_item() | Function To search item information |
| 43. | float getitemprice(char nm[]) | Function To return Item price |
| 44. | int getitemid(char nm[]) | Function To return Item ID |
| 45. | char \*getitemname(int id) | Function To return Item name |
| 46. | void deleterecord\_item() | Function To delete Item information |
| 47. | void modifyrecord\_item() | Function To modify Item information |
| 48. | int getlastcustomerid() | Function To return last customer ID |
| 49. | void showallrecord\_customer() | Function To display customer information |
| 50. | void searchrecord\_customer() | Function To search customer information |
| 51. | char \*getcustomername(int oid) | Function To return customer name |
| 52. | int getlastorderid() | Function To return last order ID |
| 53. | void showallrecord\_order() | Function To display order information |
| 54. | void searchrecord\_order() | Function To search order information |
| 55. | void showallrecord\_bill() | Function To display bill information |
| 56. | void searchrecord\_bill() | Function To search bill information |
| 57. | void get\_order() | Function To Take an order |
| 58. | void generate\_bill() | Function To Generate Bill |
| 59. | void menu\_emp() | Function of a menu to access employee information |
| 60. | void menu\_item() | Function of a menu to access item information |
| 61. | void menu\_query() | Function of a menu for any Queries |
| 62. | void menu\_report() | Function of a menu for different reports |
| 63. | void menu\_user() | Function of a menu for accessing user information |
| 64. | void menu\_main() | Function of a menu For Admin access |
| 65. | void menu\_main1() | Function of a menu for User access |
| 66. | void welcome() | Function for welcome screen |
| 67. | void password() | Function for password |
| 68. | void thank\_you() | Function for exit screen |
| 69. | void main() | Main Function |
| 70. | void addrecord\_user() | Function for adding user record |
| 71. | void showallrecord\_user() | Function to display user record |
| 72. | int getlastempno() | Function to get last employee number |

# Flow Chart

# Source Code

#include<fstream.h>

#include<time.h>

#include<string.h>

#include<conio.h>

#include<stdio.h>

#include<dos.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Structure For Getting Date

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

struct mydate

{

int dd,mm,yy;

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing employee details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class employee

{

private:

int emp\_id;

char name[20];

char sex;

char phoneno[11];

public:

void getdata\_emp(int );

void showdata\_emp(int );

int get\_empid();

void getnewdata\_emp();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing item details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class itemss

{

private:

int item\_id;

char name[20];

float price;

public:

void getdata\_item(int );

void showdata\_item(int);

void showdata\_item1(int );

int get\_itemid();

float get\_price();

char \*get\_name();

void getnewdata\_item();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing user details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class user

{

private:

char usid[20];

char pass[20];

public:

void getdata\_user();

void showdata\_user(int );

char \*get\_id();

char \*get\_ps();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing customer details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class customer

{

private:

int cust\_id;

char name[20];

char address[20];

char phone[12];

int oid;

public:

void getdata\_cust(int a,char b[],char c[],char d[],int e);

void showdata\_cust(int);

int get\_custid();

char \*get\_name();

int get\_oid();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing order details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class order

{

private:

int orno;

mydate ordate;

int itemid[5];

int qty[5];

float amt[5];

float total;

int size;

time\_t tme;

public:

int \*getitemid();

int \*getqty();

float \*getamt();

int getsize();

float gettotal();

time\_t gettt()

{

return tme;

}

mydate getdt()

{

return ordate;

}

void getdata\_order(int a,mydate b,int c[],int d[],float e[],float f,int n);

void showdata\_order(int);

int get\_orderid();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Class for accessing billing details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class bill

{

private:

int billno;

mydate dt;

char status;

float amt;

public:

void getdata\_bill(int a,mydate b,char c,float d);

void showdata\_bill(int);

int get\_billno();

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "employee"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void employee::getdata\_emp(int e)

{

emp\_id=e;

cout<<"\n Emp id "<<e;

cout<<"\n Enter name ";

gets(name);

cout<<"\n Enter sex ";

cin>>sex;

cout<<"\n Enter phone no ";

cin>>phoneno;

}

void employee::showdata\_emp(int j)

{

gotoxy(2,j);

cout<<emp\_id;

gotoxy(17,j);

cout<<name;

gotoxy(32,j);

cout<<sex;

gotoxy(42,j);

cout<<phoneno;

}

int employee::get\_empid()

{

return emp\_id;

}

void employee::getnewdata\_emp()

{

cout<<"\n Enter new phone no ";

cin>>phoneno;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "item"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void itemss::getdata\_item(int e)

{

item\_id=e;

cout<<"\n Item id "<<e;

cout<<"\n Enter name ";

gets(name);

cout<<"\n Enter price ";

cin>>price;

}

void itemss::showdata\_item(int i)

{

gotoxy(3,i);

cout<<item\_id;

gotoxy(16,i);

cout<<name;

gotoxy(29,i);

cout<<price;

}

void itemss::showdata\_item1(int r)

{

gotoxy(42,r);

cout<<name;

gotoxy(56,r);

cout<<price;

}

int itemss::get\_itemid()

{

return item\_id;

}

float itemss::get\_price()

{

return price;

}

char \*itemss::get\_name()

{

return name;

}

void itemss::getnewdata\_item()

{

cout<<"\n Enter new price ";

cin>>price;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "user"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void user::getdata\_user()

{

cout<<"\n Enter id ";

gets(usid);

cout<<"\n Enter password ";

gets(pass);

}

void user::showdata\_user(int i)

{

gotoxy(3,i);

cout<<usid;

gotoxy(16,i);

cout<<pass;

}

char \*user::get\_id()

{

return usid;

}

char \*user::get\_ps()

{

return pass;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "customer"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void customer::getdata\_cust(int a,char b[],char c[],char d[],int e)

{

cust\_id=a;

strcpy(name,b);

strcpy(address,c);

strcpy(phone,d);

oid=e;

}

void customer::showdata\_cust(int i)

{

gotoxy(3,i);

cout<<cust\_id;

gotoxy(17,i);

cout<<name;

gotoxy(30,i);

cout<<phone;

gotoxy(43,i);

cout<<address;

}

int customer::get\_custid()

{

return cust\_id;

}

char \*customer::get\_name()

{

return name;

}

int customer::get\_oid()

{

return oid;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "order"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int \*order::getitemid()

{

return itemid;

}

int \*order::getqty()

{

return qty;

}

float \*order::getamt()

{

return amt;

}

int order::getsize()

{

return size;

}

float order::gettotal()

{

return total;

}

void order::getdata\_order(int a,mydate b,int c[],int d[],float e[],float f,int n)

{

tme=time(NULL);

orno=a;

ordate=b;

size=n ;

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

{

itemid[i]=c[i];

qty[i]=d[i];

amt[i]=e[i];

}

total=f;

}

void order::showdata\_order(int j)

{

gotoxy(3,j);

cout<<orno;

gotoxy(15,j);

cout<<ordate.dd<<"/"<<ordate.mm<<"/"<<ordate.yy;

gotoxy(30,j);

cout<<total;

}

int order::get\_orderid()

{

return orno;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Functions of class "bill"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void bill::getdata\_bill(int a,mydate b,char c,float d)

{

billno=a;

dt=b;

status=c;

amt=d;

}

void bill::showdata\_bill(int i)

{

gotoxy(3,i);

cout<<billno;

gotoxy(17,i);

cout<<status;

gotoxy(29,i);

cout<<amt;

gotoxy(45,i);

cout<<dt.dd<<"/"<<dt.mm<<"/"<<dt.yy;

}

int bill::get\_billno()

{

return billno;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function for creating A Box

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void box(int r1,int c1,int r2,int c2)

{

char ch;

int i;

gotoxy(c1,r1);

ch=218;

cout<<ch;

gotoxy(c2,r1);

ch=191;

cout<<ch;

gotoxy(c1,r2);

ch=192;

cout<<ch;

gotoxy(c2,r2);

ch=217;

cout<<ch;

ch=196;

for(i=c1+1;i<c2;i++)

{

gotoxy(i,r1);

cout<<ch;

}

ch=196;

for(i=c1+1;i<c2;i++)

{

gotoxy(i,r2);

cout<<ch;

}

ch=179;

for(i=r1+1;i<r2;i++)

{

gotoxy(c1,i);

cout<<ch;

}

ch=179;

for(i=r1+1;i<r2;i++)

{

gotoxy(c2,i);

cout<<ch;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function for adding user record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void addrecord\_user()

{

clrscr();

ofstream f;

user u;

f.open("user.dat",ios::binary|ios::app);

u.getdata\_user();

f.write((char \*)&u,sizeof(u));

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to display user record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_user()

{

clrscr();

ifstream f;

user u;

int e=5;

f.open("user.dat",ios::binary);

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<"Emp\_ID";

gotoxy(15,3);

cout<<"Emp\_password";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while(f.read((char \*)&u,sizeof(u)))

{

u.showdata\_user(e);

e++;

}

f.close();

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to get last employee number

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int getlastempno()

{

ifstream f;

employee e;

int c=0;

f.open("employee.dat",ios::binary);

while(f.read((char \*)&e,sizeof(e)))

c=e.get\_empid();

f.close();

return c;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function for adding Employee record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void addrecord\_emp()

{

clrscr();

ofstream f;

employee e;

int en=getlastempno()+1;

f.open("employee.dat",ios::binary|ios::app);

e.getdata\_emp(en);

f.write((char \*)&e,sizeof(e));

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to show Employee record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_emp()

{

clrscr();

ifstream f;

employee e;

int i=5;

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<" Emp\_ID ";

gotoxy(15,3);

cout<<" Emp\_NAME ";

gotoxy(30,3);

cout<<" SEX ";

gotoxy(40,3);

cout<<" PHONE\_NO. ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

f.open("employee.dat",ios::binary);

while(f.read((char \*)&e,sizeof(e)))

{

e.showdata\_emp(i);

i++;

}

f.close();

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to search Employee information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void searchrecord\_emp()

{

clrscr();

int i=7;

ifstream f;

employee e;

int en,flag=0;

cout<<"\n Enter emp no to search ";

cin>>en;

f.open("employee.dat",ios::binary);

while(f.read((char \*)&e,sizeof(e)))

{

if(en==e.get\_empid())

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" Emp\_ID ";

gotoxy(15,5);

cout<<" Emp\_NAME ";

gotoxy(30,5);

cout<<" SEX ";

gotoxy(40,5);

cout<<" PHONE\_NO. ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

e.showdata\_emp(i);

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry empno not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To delete Employee information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void deleterecord\_emp()

{

clrscr();

ifstream f;

int i=7;

ofstream g;

employee e;

int en,flag=0;

cout<<"\n Enter emp no to delete ";

cin>>en;

f.open("employee.dat",ios::binary);

g.open("temp.dat",ios::binary);

while(f.read((char \*)&e,sizeof(e)))

{

if(en==e.get\_empid())

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" Emp\_ID ";

gotoxy(15,5);

cout<<" Emp\_NAME ";

gotoxy(30,5);

cout<<" SEX ";

gotoxy(40,5);

cout<<" PHONE\_NO. ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

e.showdata\_emp(i);

cout<<"\n Record Deleted ";

flag=1;

}

else

{

g.write((char \*)&e,sizeof(e));

}

}

f.close();

g.close();

remove("employee.dat");

rename("temp.dat","employee.dat");

if(flag==0)

{

cout<<"\n Sorry empno not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To modify Employee information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void modifyrecord\_emp()

{

clrscr();

fstream f;

employee e;

int i=7;

f.open("employee.dat",ios::binary|ios::in|ios::out);

int en,flag=0;

cout<<"\n Enter empno to modify ";

cin>>en;

while(f.read((char \*)&e,sizeof(e)))

{

if(e.get\_empid()==en)

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" Emp\_ID ";

gotoxy(15,5);

cout<<" Emp\_NAME ";

gotoxy(30,5);

cout<<" SEX ";

gotoxy(40,5);

cout<<" PHONE\_NO. ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

e.showdata\_emp(i);

e.getnewdata\_emp();

f.seekp(f.tellg()-sizeof(e),ios::beg);

f.write((char \*)&e,sizeof(e));

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry emp no not found ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To get last Item Number

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int getlastitemno()

{

ifstream f;

itemss i;

int c=0;

f.open("itemss.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

c=i.get\_itemid();

f.close();

return c;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To add item record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void addrecord\_item()

{

clrscr();

ofstream f;

itemss i;

int en=getlastitemno()+1;

f.open("itemss.dat",ios::binary|ios::app);

i.getdata\_item(en);

f.write((char \*)&i,sizeof(i));

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To display item record

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_item()

{

clrscr();

ifstream f;

itemss i;

int e=5;

f.open("itemss.dat",ios::binary);

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<"ID ";

gotoxy(15,3);

cout<<" Name ";

gotoxy(28,3);

cout<<" Price ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while(f.read((char \*)&i,sizeof(i)))

{

i.showdata\_item(e);

e++;

}

f.close();

}

void showallrecord\_item1()

{

ifstream f;

itemss i;

f.open("itemss.dat",ios::binary);

box(10,40,25,60);

int r=11;

while(f.read((char \*)&i,sizeof(i)))

{

i.showdata\_item1(r);

r++;

}

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To search item information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void searchrecord\_item()

{

clrscr();

ifstream f;

itemss i;

int a=7;

int in,flag=0;

cout<<"\n Enter item no to search ";

cin>>in;

f.open("itemss.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

{

if(in==i.get\_itemid())

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" ID ";

gotoxy(15,5);

cout<<" NAME ";

gotoxy(30,5);

cout<<" PRICE ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

i.showdata\_item(a);

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry item not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return Item price

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

float getitemprice(char nm[])

{

ifstream f;

itemss i;

float a;

f.open("itemss.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

{

if(strcmp(nm,i.get\_name())==0)

{

a=i.get\_price();

break;

}

}

f.close();

return a;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return Item ID

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int getitemid(char nm[])

{

ifstream f;

itemss i;

int id;

f.open("itemss.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

{

if(strcmp(nm,i.get\_name())==0)

{

id=i.get\_itemid();

break;

}

}

f.close();

return id;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return Item name

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \*getitemname(int id)

{

ifstream f;

itemss i;

char nm[20];

f.open("itemss.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

{

if(id==i.get\_itemid())

{

strcpy(nm,i.get\_name());

break;

}

}

f.close();

return nm;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To delete Item information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void deleterecord\_item()

{

clrscr();

ifstream f;

ofstream g;

int a=7;

itemss i;

int in,flag=0;

cout<<"\n Enter item no to delete ";

cin>>in;

f.open("itemss.dat",ios::binary);

g.open("temp.dat",ios::binary);

while(f.read((char \*)&i,sizeof(i)))

{

if(in==i.get\_itemid())

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" ID ";

gotoxy(15,5);

cout<<" NAME ";

gotoxy(30,5);

cout<<" PRICE ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

i.showdata\_item(a);

cout<<"\n Record Deleted ";

flag=1;

}

else

{

g.write((char \*)&i,sizeof(i));

}

}

f.close();

g.close();

remove("itemss.dat");

rename("temp.dat","itemss.dat");

if(flag==0)

{

cout<<"\n Sorry item not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To modify Item information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void modifyrecord\_item()

{

clrscr();

fstream f;

itemss i;

int a=7;

f.open("itemss.dat",ios::binary|ios::in|ios::out);

int in,flag=0;

cout<<"\n Enter item no to modify ";

cin>>in;

while(f.read((char \*)&i,sizeof(i)))

{

if(i.get\_itemid()==in)

{

cout<<"\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<" ID ";

gotoxy(15,5);

cout<<" NAME ";

gotoxy(30,5);

cout<<" PRICE ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

i.showdata\_item(a);

i.getnewdata\_item();

f.seekp(f.tellg()-sizeof(i),ios::beg);

f.write((char \*)&i,sizeof(i));

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry item not found ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return last customer ID

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int getlastcustomerid()

{

ifstream f;

customer cc;

int c=0;

f.open("customer.dat",ios::binary);

while(f.read((char \*)&cc,sizeof(cc)))

c=cc.get\_custid();

f.close();

return c;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To display customer information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_customer()

{

clrscr();

ifstream f;

customer cc;

int e=5;

f.open("customer.dat",ios::binary);

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<"Cutomer ID";

gotoxy(15,3);

cout<<" Name ";

gotoxy(28,3);

cout<<" Phone NO. ";

gotoxy(41,3);

cout<<" Address ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while(f.read((char \*)&cc,sizeof(cc)))

{

cc.showdata\_cust(e);

e++;

}

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To search customer information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void searchrecord\_customer()

{

clrscr();

ifstream f;

customer cc;

int ci,flag=0,e=7;

cout<<"\n Enter customer id to search ";

cin>>ci;

f.open("customer.dat",ios::binary);

while(f.read((char \*)&cc,sizeof(cc)))

{

if(ci==cc.get\_custid())

{

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(2,5);

cout<<"Cutomer ID";

gotoxy(15,5);

cout<<" Name ";

gotoxy(28,5);

cout<<" Phone NO. ";

gotoxy(41,5);

cout<<" Address ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cc.showdata\_cust(e);

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry customer not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return customer name

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \*getcustomername(int oid)

{

ifstream f;

customer cc;

char nm[20];

f.open("customer.dat",ios::binary);

while(f.read((char \*)&cc,sizeof(cc)))

{

if(oid==cc.get\_oid())

{

strcpy(nm,cc.get\_name());

break;

}

}

f.close();

return nm;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To return last order ID

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int getlastorderid()

{

ifstream f;

order oo;

int c=0;

f.open("order.dat",ios::binary);

while(f.read((char \*)&oo,sizeof(oo)))

c=oo.get\_orderid();

f.close();

return c;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To display order information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_order()

{

clrscr();

ifstream f;

order oo;

int e=5;

f.open("order.dat",ios::binary);

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<"Order NO.";

gotoxy(16,3);

cout<<"Date";

gotoxy(29,3);

cout<<"Total";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while(f.read((char \*)&oo,sizeof(oo)))

{

oo.showdata\_order(e);

e++;

}

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To search order information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void searchrecord\_order()

{

clrscr();

ifstream f;

order oo;

int oi,flag=0,e=7;

cout<<"\n Enter order id to search ";

cin>>oi;

f.open("order.dat",ios::binary);

while(f.read((char \*)&oo,sizeof(oo)))

{

if(oi==oo.get\_orderid())

{

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<"Order NO.";

gotoxy(16,5);

cout<<"Date";

gotoxy(29,5);

cout<<"Total";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

oo.showdata\_order(e);

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry order not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To display bill information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void showallrecord\_bill()

{

clrscr();

ifstream f;

bill bb;

int e=5;

f.open("bill.dat",ios::binary);

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<" Bill ID ";

gotoxy(15,3);

cout<<" Status ";

gotoxy(25,3);

cout<<" Amount ";

gotoxy(43,3);

cout<<" Date ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while(f.read((char \*)&bb,sizeof(bb)))

{

bb.showdata\_bill(e);

e++;

}

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To search bill information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void searchrecord\_bill()

{

clrscr();

ifstream f;

bill bb;

int bi,flag=0,e=7;

cout<<"\n Enter order id to search ";

cin>>bi;

f.open("order.dat",ios::binary);

while(f.read((char \*)&bb,sizeof(bb)))

{

if(bi==bb.get\_billno())

{

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

cout<<endl<<" Bill ID ";

gotoxy(15,5);

cout<<" Status ";

gotoxy(25,5);

cout<<" Amount ";

gotoxy(43,3);

cout<<" Date ";

cout<<endl<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

bb.showdata\_bill(e);

flag=1;

break;

}

}

f.close();

if(flag==0)

{

cout<<"\n Sorry bill not exist ";

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To Take an order

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void get\_order()

{

clrscr();

int oid,price[5],qty[5],id[5],i=0,q;

float amt[5],total=0;

char nm[20],ch;

struct date d;

getdate(&d);

mydate cd;

cd.yy=d.da\_year;

cd.dd=d.da\_day;

cd.mm=d.da\_mon;

showallrecord\_item1();

gotoxy(3,3);

oid=getlastorderid()+1;

cout<<" Today Date : "<<cd.dd<<"/"<<cd.mm<<"/"<<cd.yy;

gotoxy(43,3);

cout<<" Order Id "<<oid;

gotoxy(3,4);

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(4,5);

cout<<"Id";

gotoxy(10,5);

cout<<"Qty";

gotoxy(15,5);

cout<<"Price";

gotoxy(23,5);

cout<<"Amt";

gotoxy(3,6);

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

int cid=getlastcustomerid()+1 ;

char name[20],phone[20],address[20];

gotoxy(43,4);

cout<<" Cust\_id "<<cid;

gotoxy(43,5);

cout<<" Name ";

gets(name);

gotoxy(43,6);

cout<<" Phone ";

gets(phone);

gotoxy(43,7);

cout<<" Address ";

gets(address);

do

{

gotoxy(3,20);

cout<<" Enter items-: ";

gotoxy(3,21);

cout<<" Enter name ";

gets(nm);

gotoxy(3,22);

cout<<" Enter qty ";

cin>>q;

id[i]=getitemid(nm);

price[i]=getitemprice(nm);

qty[i]=q;

amt[i]=price[i]\*qty[i];

total=total+amt[i];

gotoxy(4,7+i);

cout<<id[i];

gotoxy(10,7+i);

cout<<qty[i];

gotoxy(15,7+i);

cout<<price[i];

gotoxy(23,7+i);

cout<<amt[i];

gotoxy(3,23);

cout<<" Want more ";

cin>>ch;

i++;

gotoxy(3,20);

clreol();

gotoxy(3,21);

clreol();

gotoxy(3,22);

clreol();

gotoxy(3,23);

clreol();

showallrecord\_item1();

}while(ch=='y' && i<4);

gotoxy(3,7+i);

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

gotoxy(4,8+i);

cout<<" Total Bill "<<total;

getch();

customer cc;

cc.getdata\_cust(cid,name,address,phone,oid);

ofstream fc;

fc.open("customer.dat",ios::binary|ios::app);

fc.write((char \*)&cc,sizeof(cc));

fc.close();

order oo;

oo.getdata\_order(oid,cd,id,qty,amt,total,i);

ofstream fo;

fo.open("order.dat",ios::binary|ios::app);

fo.write((char \*)&oo,sizeof(oo));

fo.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function To Generate Bill

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void generate\_bill()

{

clrscr();

ifstream f;

order oo;

bill bb;

time\_t ot,ct;

ct=time(NULL);

int oi,diff,n;

char st;

float a;

mydate dt;

cout<<"\n Enter order id ";

cin>>oi;

f.open("order.dat",ios::binary);

while(f.read((char \*)&oo,sizeof(oo)))

{

if(oi==oo.get\_orderid())

{

ot=oo.gettt();

dt=oo.getdt();

a=oo.gettotal();

diff=difftime(ct,ot);

if(diff>300)

{

a=a-a\*15/100;

st='d';

}

else

{

st='o';

}

box(4,4,20,50);

struct date d;

getdate(&d);

mydate cd;

cd.yy=d.da\_year;

cd.dd=d.da\_day;

cd.mm=d.da\_mon;

int r=5,i;

n=oo.getsize();

gotoxy(5,r);

cout<<" Today Date : "<<cd.dd<<"/"<<cd.mm<<"/"<<cd.yy;

gotoxy(35,r);

cout<<" Order Id "<<oi;

r++;

gotoxy(5,r);

cout<<" Name "<<getcustomername(oi);

r++;

for(i=5;i<=49;i++)

{

gotoxy(i,r);

cout<<(char)196;

}

r++;

gotoxy(5,r);

cout<<" Item Name ";

gotoxy(20,r);

cout<<"Qty";

gotoxy(25,r);

cout<<"Price";

gotoxy(33,r);

cout<<"Amount";

r++;

for(i=5;i<=49;i++)

{

gotoxy(i,r);

cout<<(char)196;

}

int \*id,\*qt;

float \*am;

r++;

id=oo.getitemid();

qt=oo.getqty();

am=oo.getamt();

for(i=0;i<n;i++)

{

gotoxy(6,r);

cout<<getitemname(\*id);

gotoxy(20,r);

cout<<\*qt;

gotoxy(25,r);

cout<<(\*am)/(\*qt);

gotoxy(33,r);

cout<<\*am;

id++;

qt++;

am++;

r++;

}

for(i=5;i<=49;i++)

{

gotoxy(i,r);

cout<<(char)196;

}

r++;

gotoxy(25,r);

cout<<" Total "<<oo.gettotal();

r++;

gotoxy(25,r) ;

if(st=='d')

cout<<" Discount "<<oo.gettotal()-a;

else

cout<<" Discount "<<0;

r++;

gotoxy(25,r);

cout<<" Net Amt "<<a;

bb.getdata\_bill(oi,dt,st,a);

ofstream fb;

fb.open("bill.dat",ios::binary|ios::app);

fb.write((char \*)&bb,sizeof(bb));

fb.close();

break;

}

}

f.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu to access employee information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_emp()

{

char op;

do

{

clrscr();

box(3,13,13,50);

gotoxy(14,4);

cout<<" 1 Add";

gotoxy(14,5);

cout<<" 2 Modify ";

gotoxy(14,6);

cout<<" 3 Delete ";

gotoxy(14,7);

cout<<" 0 Back ";

gotoxy(14,8);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

addrecord\_emp();

break;

case '2':

modifyrecord\_emp();

break;

case '3':

deleterecord\_emp();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu to access item information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_item()

{

char op;

do

{

clrscr();

box(3,13,13,50);

gotoxy(14,4);

cout<<" 1 Add";

gotoxy(14,5);

cout<<" 2 Modify ";

gotoxy(14,6);

cout<<" 3 Delete ";

gotoxy(14,7);

cout<<" 0 Back ";

gotoxy(14,8);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

addrecord\_item();

break;

case '2':

modifyrecord\_item();

break;

case '3':

deleterecord\_item();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu for any Queries

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_query()

{

char op;

do

{

clrscr();

box(3,13,15,50);

gotoxy(14,4);

cout<<" 1 Employee ";

gotoxy(14,5);

cout<<" 2 Customer ";

gotoxy(14,6);

cout<<" 3 Item ";

gotoxy(14,7);

cout<<" 4 Order ";

gotoxy(14,8);

cout<<" 5 Bill ";

gotoxy(14,9);

cout<<" 0 Back ";

gotoxy(14,10);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

searchrecord\_emp();

break;

case '2':

searchrecord\_customer();

break;

case '3':

searchrecord\_item();

break;

case '4':

searchrecord\_order();

break;

case '5':

searchrecord\_bill();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu for different reports

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_report()

{

char op;

do

{

clrscr();

box(3,13,15,50);

gotoxy(14,4);

cout<<" 1 Employee ";

gotoxy(14,5);

cout<<" 2 Customer ";

gotoxy(14,6);

cout<<" 3 Item ";

gotoxy(14,7);

cout<<" 4 Order ";

gotoxy(14,8);

cout<<" 5 Bill ";

gotoxy(14,9);

cout<<" 0 Back ";

gotoxy(14,10);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

showallrecord\_emp();

break;

case '2':

showallrecord\_customer();

break;

case '3':

showallrecord\_item();

break;

case '4':

showallrecord\_order();

break;

case '5':

showallrecord\_bill();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu for accessing user information

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_user()

{

char op;

do

{

clrscr();

box(3,13,11,50);

gotoxy(14,4);

cout<<" 1 Add ";

gotoxy(14,5);

cout<<" 2 Show ";

gotoxy(14,6);

cout<<" 0 Back ";

gotoxy(14,7);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

addrecord\_user();

break;

case '2':

showallrecord\_user();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu For Admin access

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_main()

{

char op;

do

{

clrscr();

box(3,13,17,50);

gotoxy(14,4);

cout<<" 1 Emp(Add/Modify/Delete) ";

gotoxy(14,5);

cout<<" 2 Item(Add/Modify/Delete) ";

gotoxy(14,6);

cout<<" 3 Order ";

gotoxy(14,7);

cout<<" 4 Generate Bill ";

gotoxy(14,8);

cout<<" 5 Query ";

gotoxy(14,9);

cout<<" 6 Report ";

gotoxy(14,10);

cout<<" 7 Create user ";

gotoxy(14,11);

cout<<" 0 Exit ";

gotoxy(14,12);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

menu\_emp();

break;

case '2':

menu\_item();

break;

case '3':

get\_order();

break;

case '4':

generate\_bill();

break;

case '5':

menu\_query();

break;

case '6':

menu\_report();

break;

case '7':

menu\_user();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function of a menu for User access

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void menu\_main1()

{

char op;

do

{

clrscr();

box(3,13,13,50);

gotoxy(14,4);

cout<<" 1 Order ";

gotoxy(14,5);

cout<<" 2 Generate Bill ";

gotoxy(14,6);

cout<<" 3 Query ";

gotoxy(14,7);

cout<<" 4 Report ";

gotoxy(14,8);

cout<<" 0 Exit ";

gotoxy(14,9);

cout<<" Enter choice ";

op=getche();

switch(op)

{

case '1':

get\_order();

break;

case '2':

generate\_bill();

break;

case '3':

menu\_query();

break;

case '4':

menu\_report();

break;

}

getch();

}while(op!='0');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function for Welcome Screen

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void welcome()

{

clrscr();

box(3,3,20,80);

box(5,19,11,55);

gotoxy(23,7);

cout<<"WELCOME";

gotoxy(33,8);

cout<<"TO";

gotoxy(40,9);

cout<<"SNACKS PARLOUR ";

for(int i=4;i<80;i++)

{

gotoxy(i,18);

cout<<(char)196;

}

gotoxy(50,15);

cout<<"Made By : -";

gotoxy(63,16) ;

cout<<"Paarth Sethi ";

gotoxy(63,17);

cout<<"XII A";

gotoxy(45,19);

cout<<"Press any key to continue.....";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function For Password

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void password()

{

char id[20],ps[20],ch;

int i=0,flag=0;

clrscr();

box(6,13,13,53);

gotoxy(15,8);

cout<<"Enter id ";

gotoxy(35,8);

gets(id);

gotoxy(15,9);

cout<<"Enter pasword ";

gotoxy(35,9);

while(1)

{

ch=getch();

if(ch==13)

break;

else if(ch==8)

{

if(i>0)

{

i--;

gotoxy(35+i,9);

clreol();

box(6,13,13,53);

gotoxy(35+i,9);

}

}

else

{

cout<<"\*";

ps[i++]=ch;

}

}

ps[i]='\0';

if ( (strcmp(id,"admin")==0) && (strcmp(ps,"admin")==0) )

menu\_main();

else

{

ifstream f;

user u;

f.open("user.dat",ios::binary);

while(f.read((char \*)&u,sizeof(u)))

{

if ( (strcmp(id,u.get\_id())==0) && (strcmp(ps,u.get\_ps())==0) )

{

flag=1;

break;

}

}

f.close();

if(flag==1)

menu\_main1();

else

{

gotoxy(15,11);

cout<<"Sorry Invalid Login";

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function for Exit Screen

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void thank\_you()

{

clrscr();

box(3,3,20,80);

box(5,19,11,55);

gotoxy(23,7);

cout<<"THANK YOU";

gotoxy(31,8);

cout<<"FOR COMING TO ";

gotoxy(40,9);

cout<<"SNACKS PARLOUR ";

for(int i=4;i<80;i++)

{

gotoxy(i,18);

cout<<(char)196;

}

gotoxy(50,15);

cout<<"Made By : -";

gotoxy(63,16) ;

cout<<"Paarth Sethi ";

gotoxy(63,17);

cout<<"XII A";

gotoxy(45,19);

cout<<"Press any key to exit.....";

}

void main()

{

clrscr();

welcome();

getch();

password();

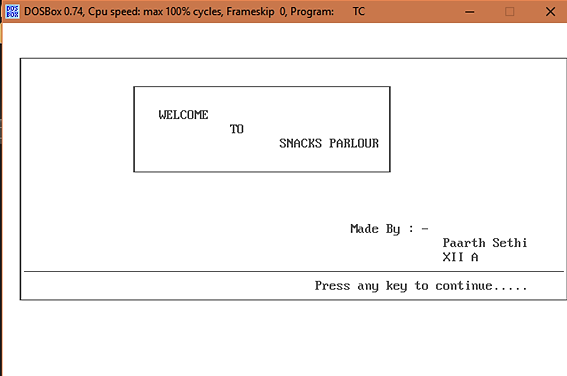
getch();

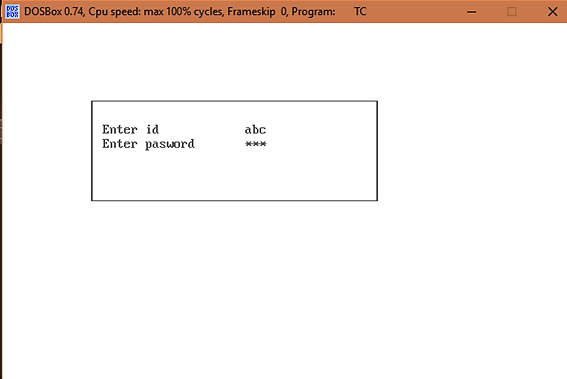
thank\_you();

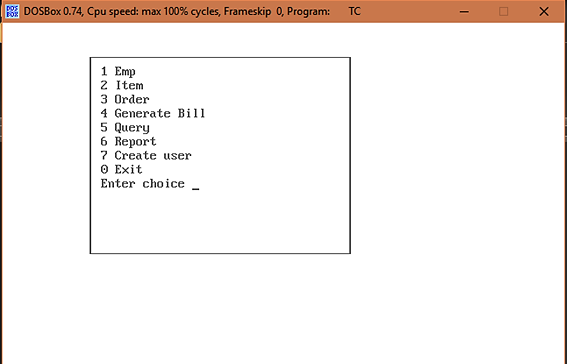
getch();

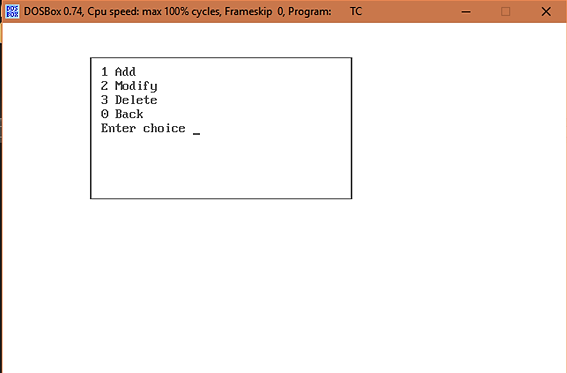
}

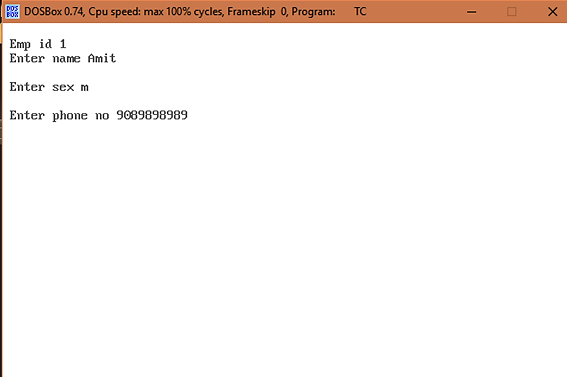
# Outputs

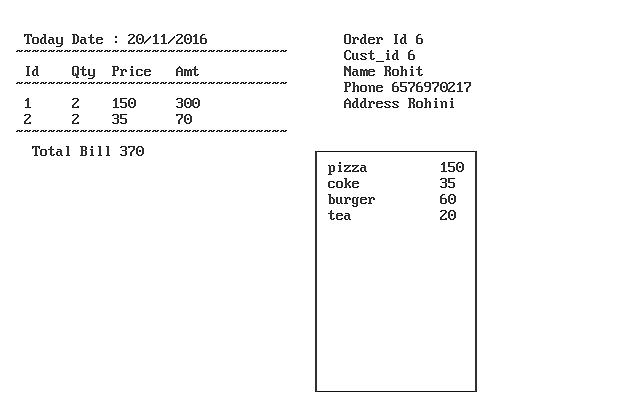


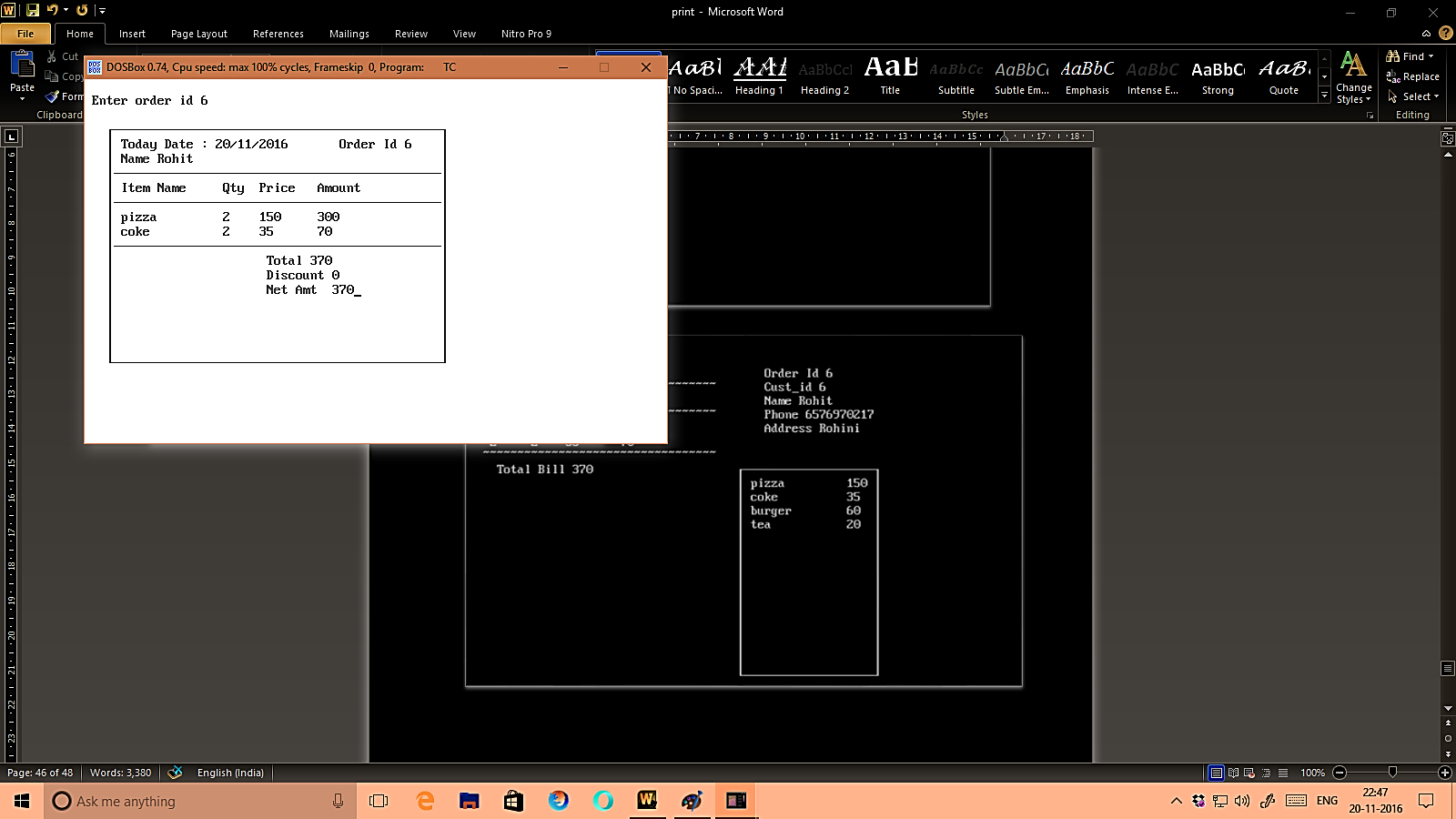


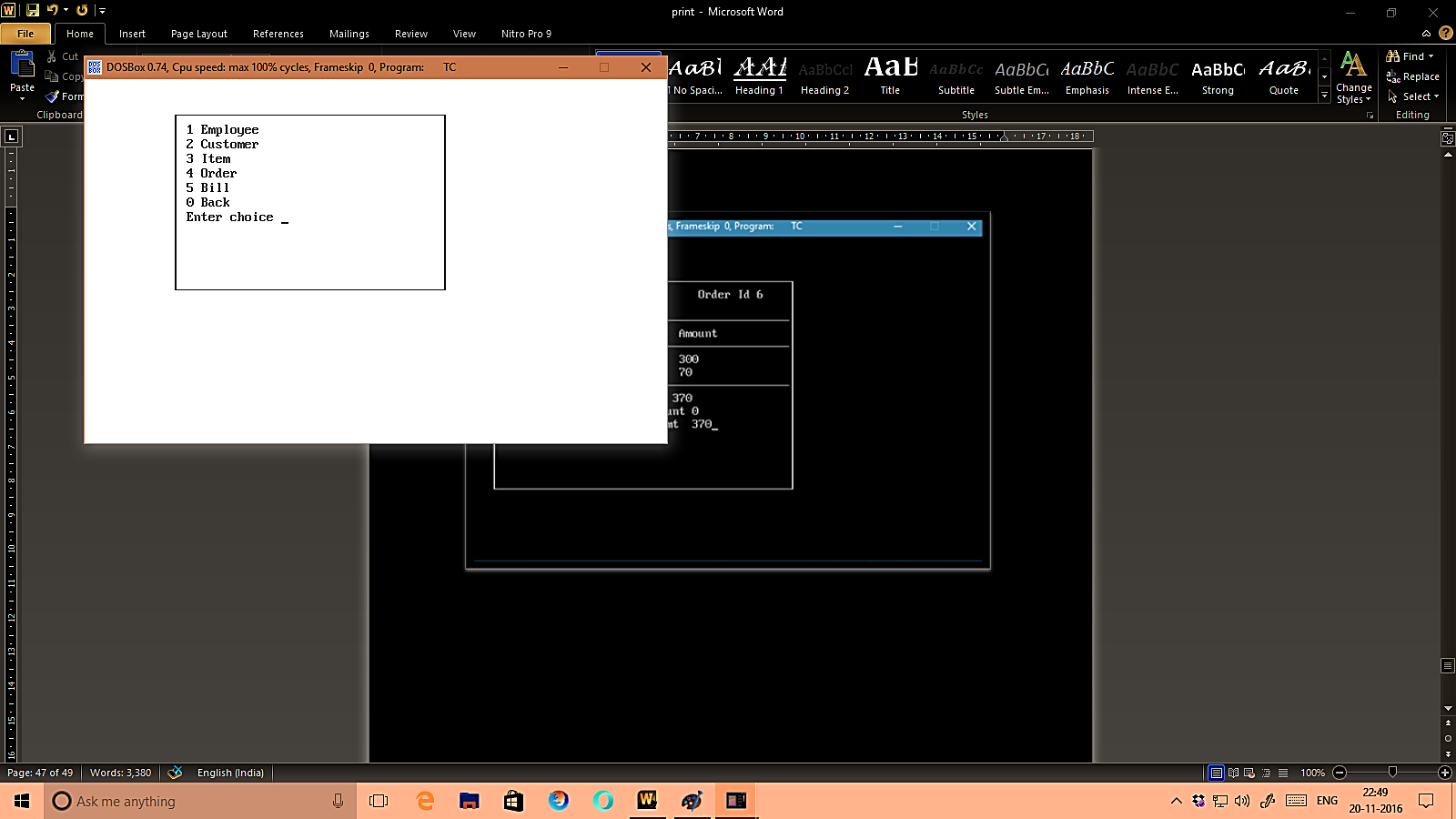


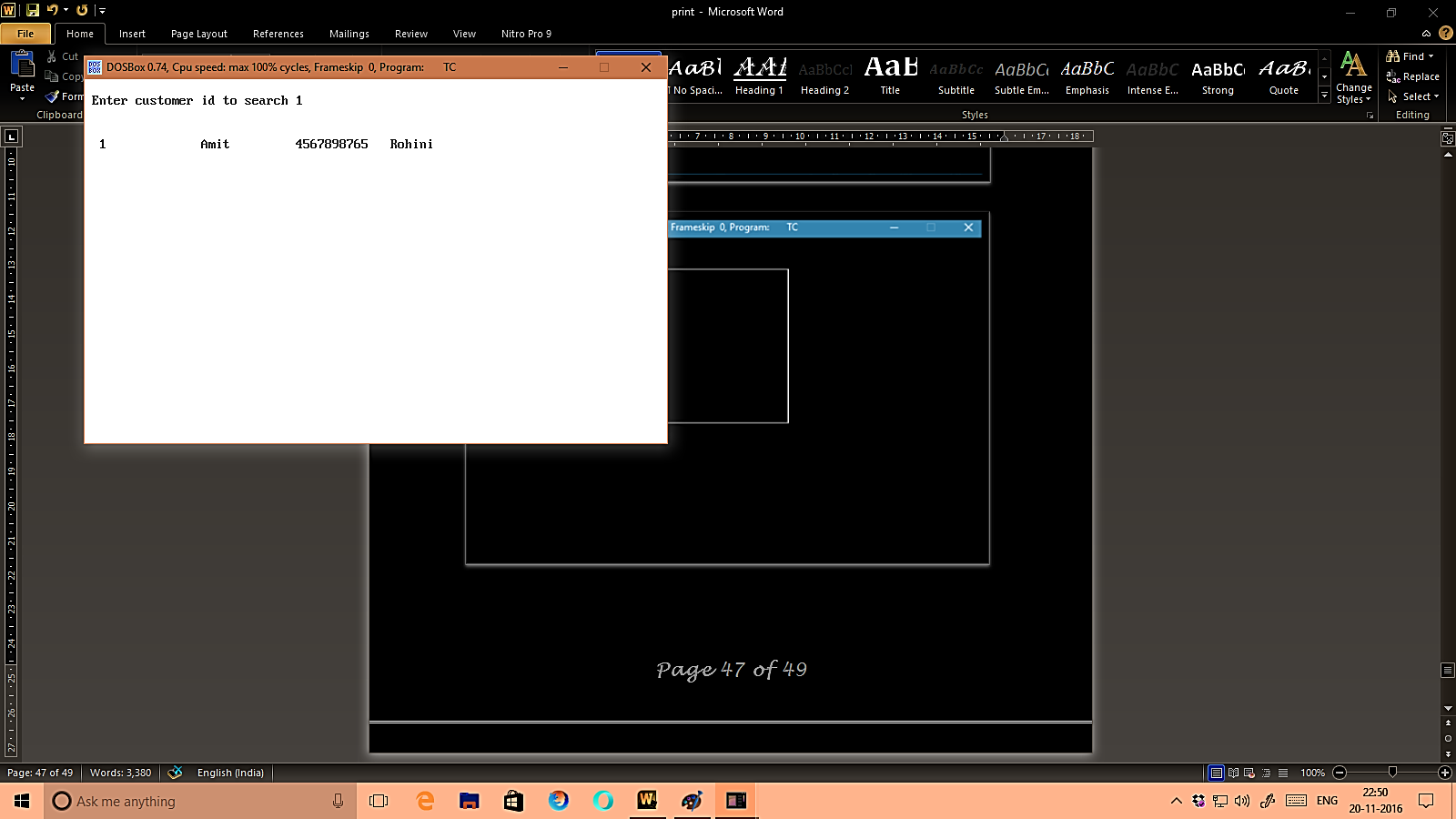


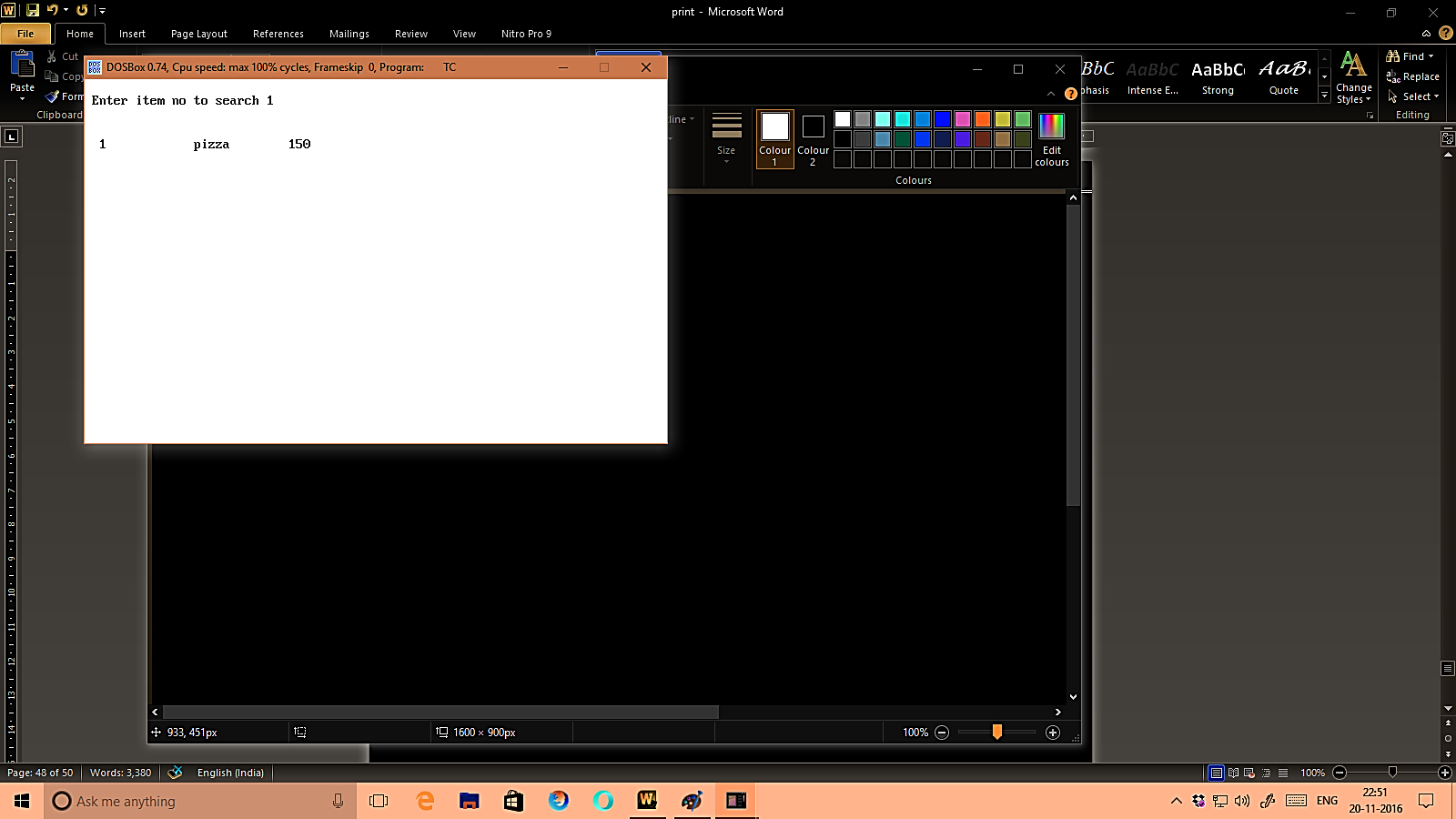


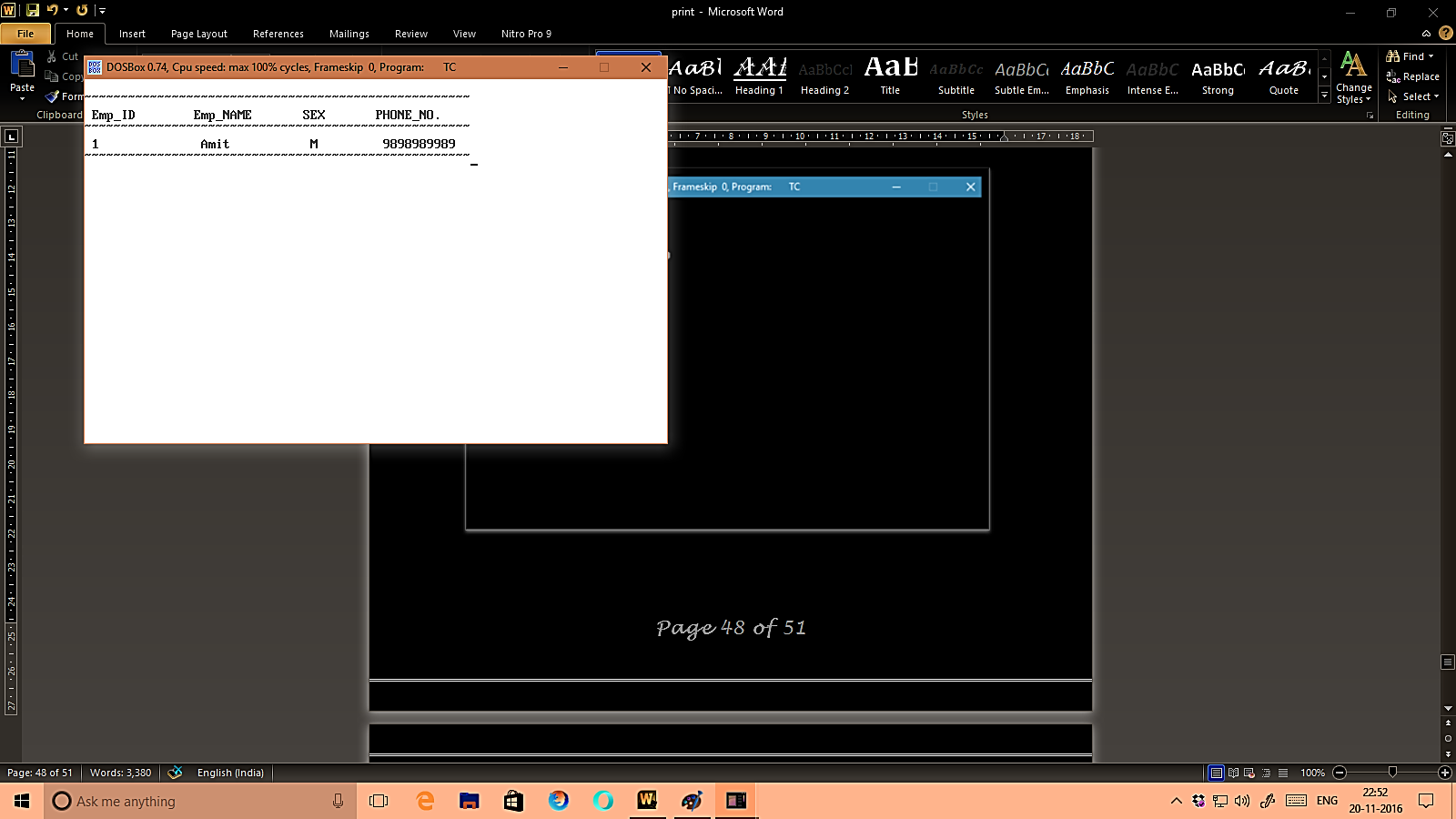


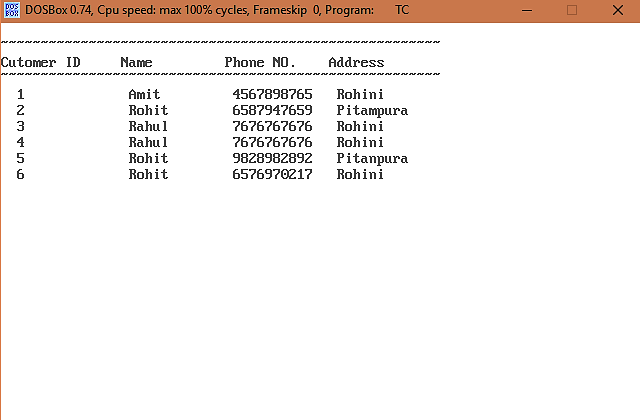


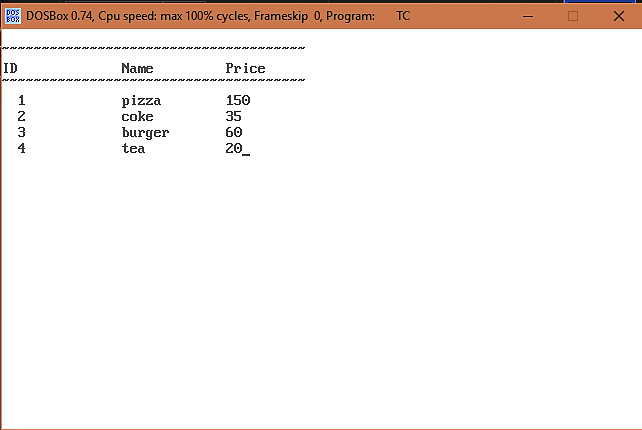


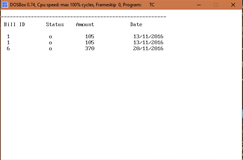


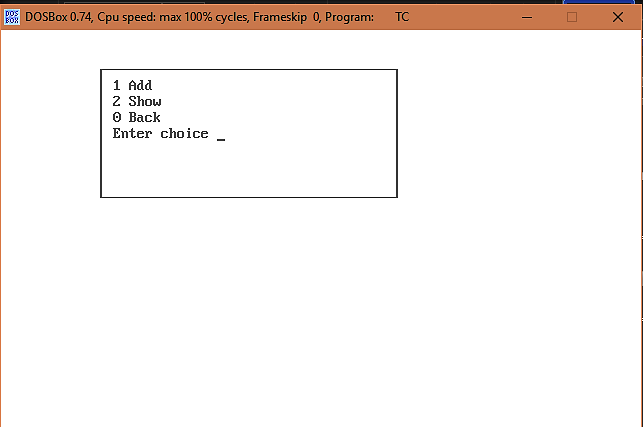


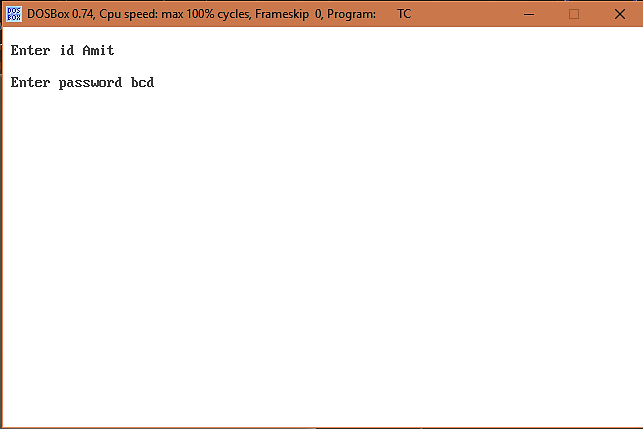


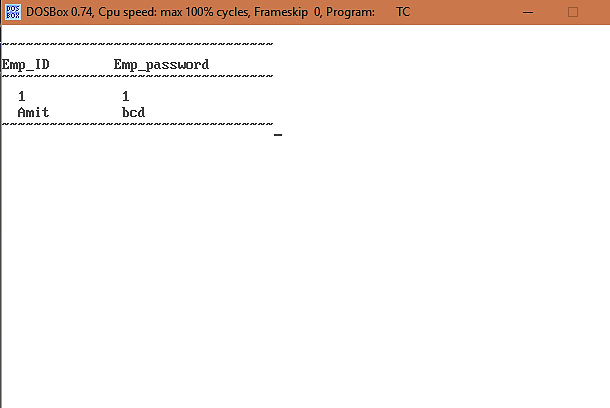


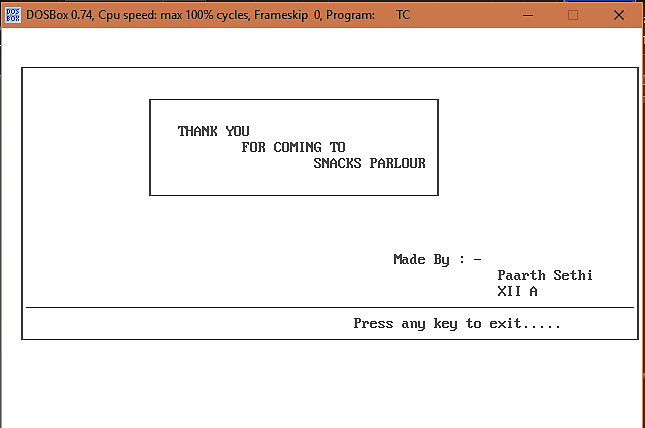












# Limitations

* No kind of checking of information is done all the data entered is assumed to be corrected.
* Admin ID and Password cannot be changed.
* Scope is limited.
* Platform Dependent.
* Not applicable in web environment.

# Bibliography

* website : [www.cppforschool.com](http://www.cppforschool.com/projects.html)
* Arora, P., Gupta, P. (2016) “Computer Science with C++ (Volume I)” New Delhi: Sultan Chand and Sons (P) Ltd.
* Arora, P., Gupta, P. (2016) “Computer Science with C++ (Volume II)” New Delhi: Sultan Chand and Sons (P) Ltd.