

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **B.Tech** | Semester | : | **Winter-21** |
| Course | : | **Database Management Systems (Embedded Lab)** | Code | : | **CSE2004** |
| Faculty | : | **Dr.V.Vani** | Slot | : | **D1** |

**TELEPHONE BILLING SYSTEM**

**Name: RIDDHIMAN RAGURAMAN**

**Reg.no: 20BCE1624**

**Subject: Database management system project report.**

**Overview**

* Abstract
* Why telephone billing system?
* Existing system
* Limitations of existing system
* Proposed system
* How it works?
* System architecture
* Er diagram and relational schema
* Attributes and dataset
* Code
* Output
* Conclusion
* **Abstract:**

A telephone billing system is a computerized system used by a business that will automatically calculate and produce customer’s pay bills. It will also view and update customers details and also delete them using suitable commands. Also, there is a separate database where the billing details like number of calls, SMS, MMS etc are stored. There is also a third database which stores the phone numbers of the customers and another data base which have the customer’s bank account number and their mode of payment of their bill.

In all these databases we have the facility to add, edit, delete and view the details. All these 4 databases are connected by foreign keys and each database have a suitable primary key.

* **Software used?**
* Visual C++ 11
* Oracle database 11g standard edition
* Oracle pre compiler
* **Why telephone billing system?**

To create a telephone billing system used by admin which is user friendly for the users to use and calculate the bill amounts.

* **Existing system:**

Many existing systems in this model uses binary data file as it is very much compatible with C++.

* **Limitations of existing system:**

Existing file for this billing system uses binary data file so

by virtue, and flaws of it when the user exists the program

the file gets removed and the user cannot access the file

again.

* **Proposed system:**

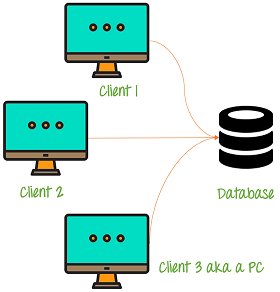
Our system uses oracle database as backend and C++ as front end for the program.

* **How it works?**

First it shows an admin homepage where u need to put the username and password to access our databases.

Then a menu would be displayed of various databases and to view the tariff on how we calculate the telephone bills and a choice to exit the program. Inside each of the options you would see another set of options which are add, edit, delete, view and return to the main menu. Exception is the phone database where u can’t edit the phone number.

* **System architecture:**

****

It is basic client-server. But Applications on the client end can directly communicate with the database at the server side.

For this interaction, API’s like: ODBC, JDBC are used. The user interfaces and application programs are run on the client-side.

The server side is responsible to provide the functionalities like: query processing and transaction management.

To communicate with the DBMS, client-side application establishes a connection with the server side.

In this software we use Oracle as our backend language and C++ as our frontend

We use C++ to create a client user face and the data is stored in DBMS which is oracle.

* **How it works?**

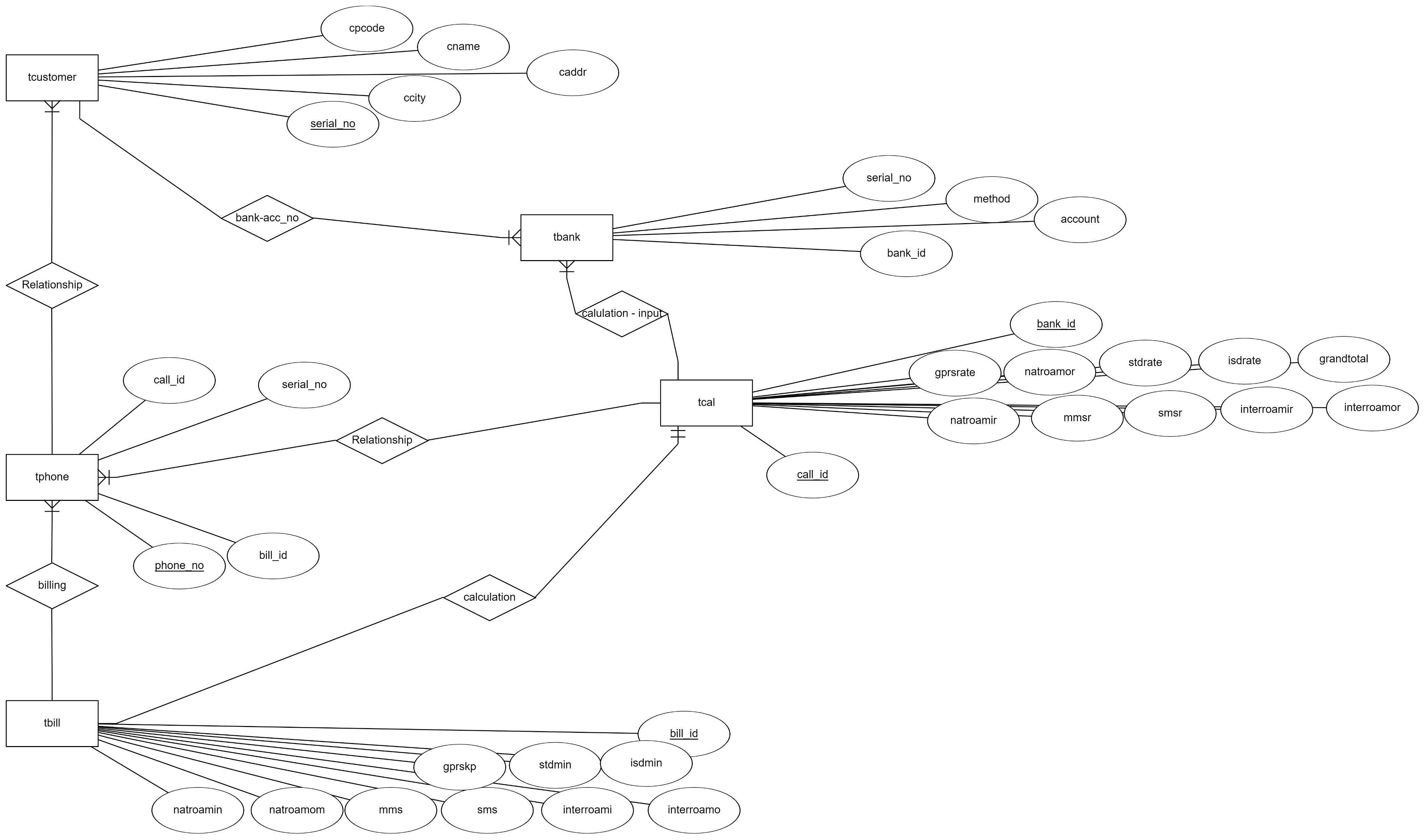
First it shows an admin homepage where u need to put the username and password to access our databases.

Then a menu would be displayed of various databases and to view the tariff on how we calculate the telephone bills and a choice to exit the program. Inside each of the options you would see another set of options which are add, edit, delete, view and return to the main menu. Exception is the phone database where u can’t edit the phone number.

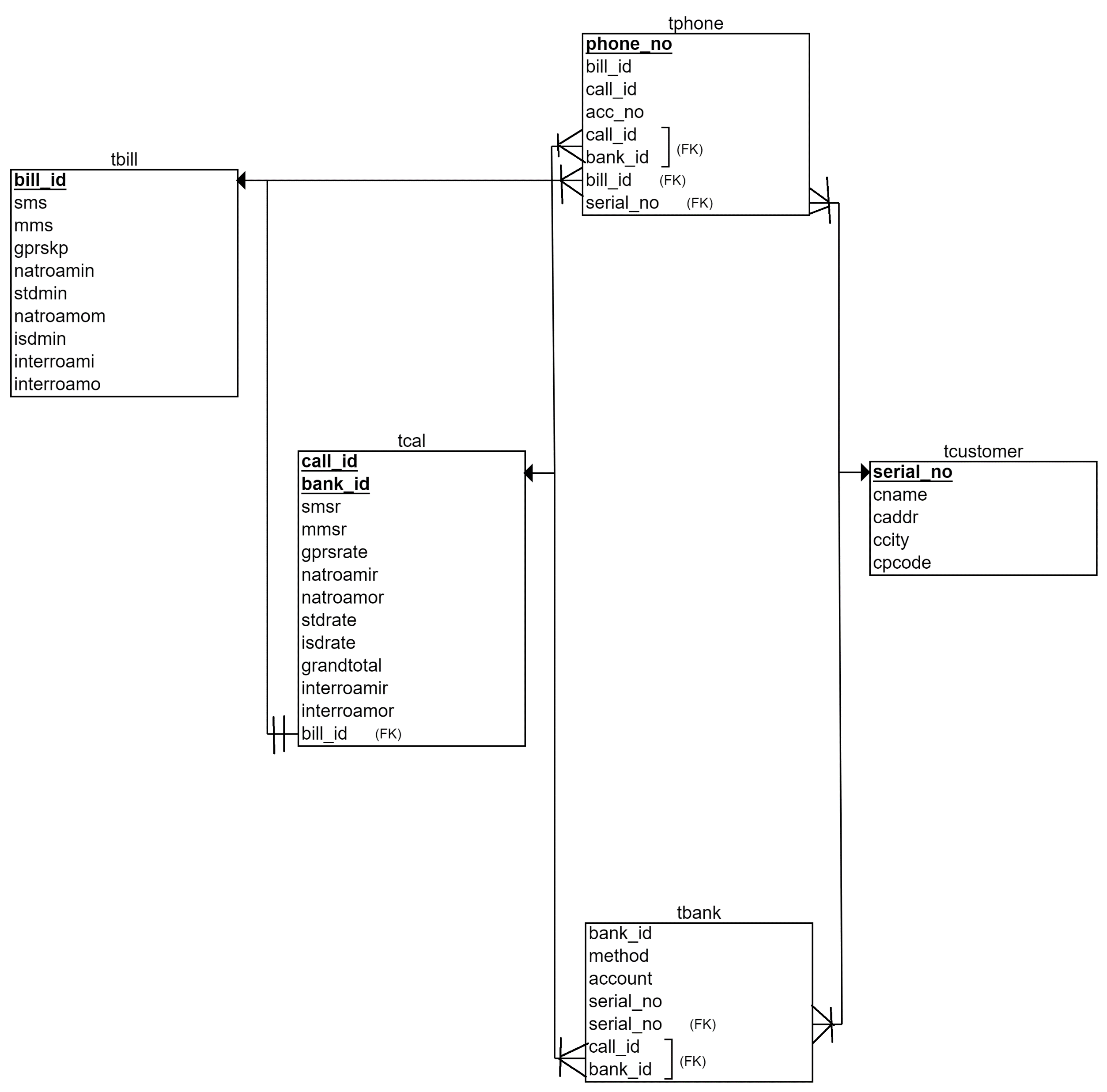
The usage of oracle in our C++ is done by writing a proc program which is integrated C and PL/SQL program which is then runed in command prompt where the installed oracle system variable with oracle pre complier converts the given ‘.pc’ file (proc) into C or C++ file.

This ‘.pc’ generates C++ file only if you have it in a same project file ‘.sln’ which should be an C++ project in turn, this ‘.pc’ file is not executed by any compliers or interpreters it is a program run in system environment so the project file must include the provided msvc ‘.dll’ file from pre compiler file in the oracle home to provide the suitable loading space for the project, the file for oracle 11g standard edition is orasqxll.dll which should be added as resource file in the project

* **Er diagram and relational schema:**
* **ER Diagram:**

****

* **Relational schema:**

****

* **Attributes and dataset:**
* **Table TCUSTOMER:**

1. ACC\_NO- which acts as a primary key which is unique for all customers.
2. CNAME- accepts the name of the customer
3. CADDR- residential address of the customer
4. CCITY- city where the customer lives in
5. CPCODE- pin code of the area

* **Table TBILL:**

1. BILL\_ID- primary key which is unique for every bill.
2. SMS- number of messages
3. MMS- number of mms
4. GPRSKP- GPRS in kb
5. STDMIN- std minutes
6. NATROAMIM- national roaming incoming minutes
7. NATROAMOM- national roaming outgoing minutes
8. ISDMIN- isd minutes
9. INTERROAMI- international roaming incoming minutes

10.INTERROAMO- international roaming outgoing minutes

* **Table TCAL:**

1. CALL\_ID-primary key for this table.
2. BANK\_ID- foreign key from the table Tbill
3. SMSR- SMS rate
4. MMSR- MMS rate
5. GPRSRATE- GPRS rate
6. NATROAMIR- national roaming incoming minutes rate
7. NATROAMOR- national roaming outgoing minutes rate
8. STDRATE- std rate
9. INTERROAMIR- international roaming incoming minutes rate

10. INTERROAMOR- international roaming outgoing minutes rate

11. ISDRATE- ISD rate

12. GRANDTOTAL- Grand Total of the bill.

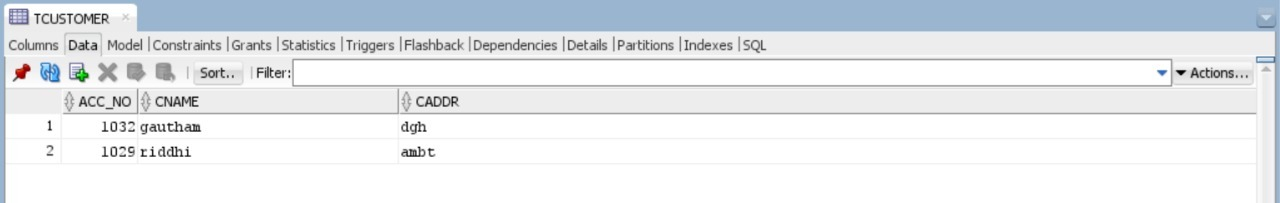
* **Table TPHONE:**

1. PHONE NO- primary key which accepts phone number for the customer
2. BILL\_ID- foreign key from the Tbill
3. CALL\_ID- foreign key from the Tcal
4. ACC\_NO- foreign key from Tcustomer

* **Table BANK:**

1. BANK\_ID- primary key for this table
2. ACCOUNT- unique key which is the bank account number of the customer
3. METHOD- which stores the mode of payment
4. ACC NO- foreign key from Tcustomer

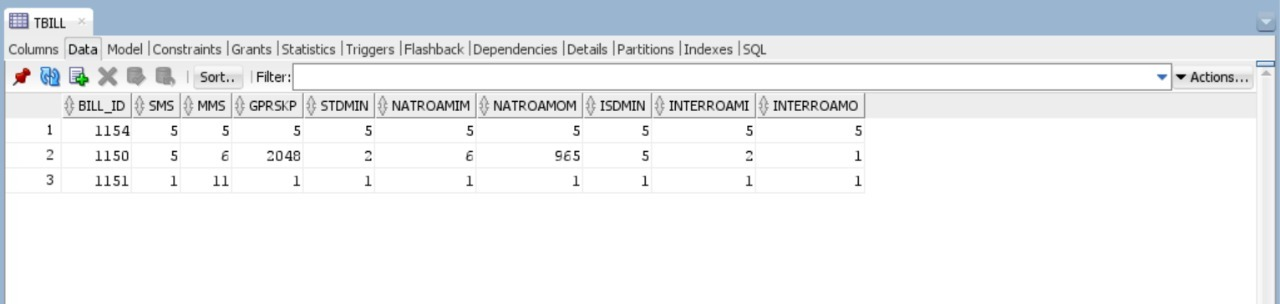
* **Sample dataset:**
  + TCUSTOMER



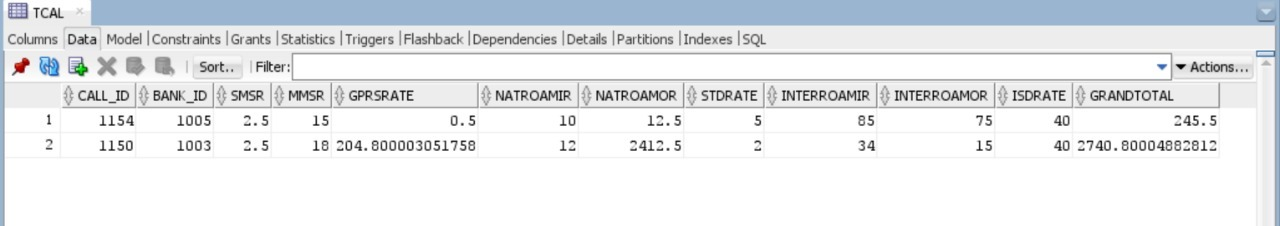
* TPHONE



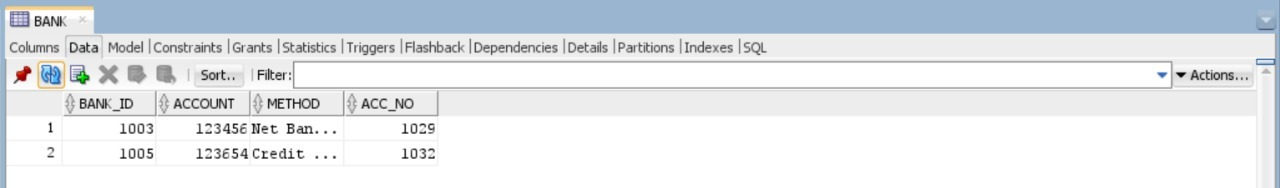
* TBILL



* TCAL



* BANK



* **Code:**

/\* Result Sets Interface \*/

#ifndef SQL\_CRSR

#  define SQL\_CRSR

  struct sql\_cursor

  {

    unsigned int curocn;

    void \*ptr1;

    void \*ptr2;

    unsigned int magic;

  };

  typedef struct sql\_cursor sql\_cursor;

  typedef struct sql\_cursor SQL\_CURSOR;

#endif /\* SQL\_CRSR \*/

/\* Thread Safety \*/

typedef void \* sql\_context;

typedef void \* SQL\_CONTEXT;

/\* Object support \*/

struct sqltvn

{

  unsigned char \*tvnvsn;

  unsigned short tvnvsnl;

  unsigned char \*tvnnm;

  unsigned short tvnnml;

  unsigned char \*tvnsnm;

  unsigned short tvnsnml;

};

typedef struct sqltvn sqltvn;

struct sqladts

{

  unsigned int adtvsn;

  unsigned short adtmode;

  unsigned short adtnum;

  sqltvn adttvn[1];

};

typedef struct sqladts sqladts;

static struct sqladts sqladt = {

  1,1,0,

};

/\* Binding to PL/SQL Records \*/

struct sqltdss

{

  unsigned int tdsvsn;

  unsigned short tdsnum;

  unsigned char \*tdsval[1];

};

typedef struct sqltdss sqltdss;

static struct sqltdss sqltds =

{

  1,

  0,

};

/\* File name & Package Name \*/

struct sqlcxp

{

  unsigned short fillen;

           char  filnam[10];

};

static const struct sqlcxp sqlfpn =

{

    9,

    "source.pc"

};

static unsigned int sqlctx = 41459;

static struct sqlexd {

   unsigned int   sqlvsn;

   unsigned int   arrsiz;

   unsigned int   iters;

   unsigned int   offset;

   unsigned short selerr;

   unsigned short sqlety;

   unsigned int   occurs;

      const short \*cud;

   unsigned char  \*sqlest;

      const char  \*stmt;

   sqladts \*sqladtp;

   sqltdss \*sqltdsp;

            void  \*\*sqphsv;

   unsigned int   \*sqphsl;

            int   \*sqphss;

            void  \*\*sqpind;

            int   \*sqpins;

   unsigned int   \*sqparm;

   unsigned int   \*\*sqparc;

   unsigned short  \*sqpadto;

   unsigned short  \*sqptdso;

   unsigned int   sqlcmax;

   unsigned int   sqlcmin;

   unsigned int   sqlcincr;

   unsigned int   sqlctimeout;

   unsigned int   sqlcnowait;

              int   sqfoff;

   unsigned int   sqcmod;

   unsigned int   sqfmod;

            void  \*sqhstv[12];

   unsigned int   sqhstl[12];

            int   sqhsts[12];

            void  \*sqindv[12];

            int   sqinds[12];

   unsigned int   sqharm[12];

   unsigned int   \*sqharc[12];

   unsigned short  sqadto[12];

   unsigned short  sqtdso[12];

} sqlstm = {12,12};

// Prototypes

extern "C" {

  void sqlcxt (void \*\*, unsigned int \*,

               struct sqlexd \*, const struct sqlcxp \*);

  void sqlcx2t(void \*\*, unsigned int \*,

               struct sqlexd \*, const struct sqlcxp \*);

  void sqlbuft(void \*\*, char \*);

  void sqlgs2t(void \*\*, char \*);

  void sqlorat(void \*\*, unsigned int \*, void \*);

}

// Forms Interface

static const int IAPSUCC = 0;

static const int IAPFAIL = 1403;

static const int IAPFTL  = 535;

extern "C" { void sqliem(unsigned char \*, signed int \*); }

 static const char \*sq0016 =

"select phone\_no  from tphone where acc\_no=:b0           ";

typedef struct { unsigned short len; unsigned char arr[1]; } VARCHAR;

typedef struct { unsigned short len; unsigned char arr[1]; } varchar;

/\* cud (compilation unit data) array \*/

static const short sqlcud0[] =

{12,4130,178,0,0,

5,0,0,1,85,0,4,51,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

36,0,0,2,52,0,4,88,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

59,0,0,3,73,0,3,89,0,0,4,4,0,1,0,1,3,0,0,1,3,0,0,1,97,0,0,1,3,0,0,

90,0,0,4,85,0,4,101,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

121,0,0,5,52,0,4,104,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

144,0,0,6,76,0,4,107,0,0,4,1,0,1,0,2,3,0,0,2,97,0,0,2,3,0,0,1,3,0,0,

175,0,0,7,88,0,4,108,0,0,5,1,0,1,0,2,97,0,0,2,97,0,0,2,97,0,0,2,3,0,0,1,3,0,0,

210,0,0,8,85,0,4,127,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

241,0,0,9,74,0,5,180,0,0,4,4,0,1,0,1,3,0,0,1,97,0,0,1,3,0,0,1,3,0,0,

272,0,0,10,0,0,29,188,0,0,0,0,0,1,0,

287,0,0,11,88,0,4,205,0,0,5,1,0,1,0,2,97,0,0,2,97,0,0,2,97,0,0,2,3,0,0,1,3,0,0,

322,0,0,12,44,0,4,209,0,0,1,0,0,1,0,2,3,0,0,

341,0,0,13,44,0,4,210,0,0,1,0,0,1,0,2,3,0,0,

360,0,0,14,77,0,3,211,0,0,4,4,0,1,0,1,97,0,0,1,3,0,0,1,3,0,0,1,3,0,0,

391,0,0,15,0,0,29,218,0,0,0,0,0,1,0,

406,0,0,16,56,0,9,235,0,0,1,1,0,1,0,1,3,0,0,

425,0,0,16,0,0,13,242,0,0,1,0,0,1,0,2,97,0,0,

444,0,0,16,0,0,15,246,0,0,0,0,0,1,0,

459,0,0,17,88,0,4,254,0,0,5,1,0,1,0,2,97,0,0,2,97,0,0,2,97,0,0,2,3,0,0,1,3,0,0,

494,0,0,18,43,0,4,276,0,0,1,0,0,1,0,2,3,0,0,

513,0,0,19,84,0,3,277,0,0,5,5,0,1,0,1,3,0,0,1,97,0,0,1,97,0,0,1,97,0,0,1,3,0,0,

548,0,0,20,0,0,29,286,0,0,0,0,0,1,0,

563,0,0,21,88,0,4,297,0,0,5,1,0,1,0,2,97,0,0,2,97,0,0,2,97,0,0,2,3,0,0,1,3,0,0,

598,0,0,22,79,0,5,332,0,0,5,5,0,1,0,1,97,0,0,1,97,0,0,1,97,0,0,1,3,0,0,1,3,0,0,

633,0,0,23,0,0,29,340,0,0,0,0,0,1,0,

648,0,0,24,44,0,4,362,0,0,1,0,0,1,0,2,3,0,0,

667,0,0,25,85,0,4,363,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

698,0,0,26,147,0,3,385,0,0,10,10,0,1,0,1,3,0,0,1,3,0,0,1,4,0,0,1,4,0,0,1,4,0,0,

1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,3,0,0,

753,0,0,27,183,0,3,386,0,0,12,12,0,1,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,

1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,3,0,0,1,3,0,0,

816,0,0,28,0,0,29,391,0,0,0,0,0,1,0,

831,0,0,29,85,0,4,407,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

862,0,0,30,156,0,4,408,0,0,10,1,0,1,0,2,3,0,0,2,3,0,0,2,4,0,0,2,4,0,0,2,4,0,0,

2,4,0,0,2,4,0,0,2,4,0,0,2,4,0,0,1,3,0,0,

917,0,0,31,180,0,4,409,0,0,11,1,0,1,0,2,4,0,0,2,4,0,0,2,4,0,0,2,4,0,0,2,4,0,0,

2,4,0,0,2,4,0,0,2,4,0,0,2,4,0,0,2,4,0,0,1,3,0,0,

976,0,0,32,85,0,4,540,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,97,0,0,1,3,0,0,

1007,0,0,33,142,0,5,541,0,0,10,10,0,1,0,1,3,0,0,1,3,0,0,1,4,0,0,1,4,0,0,1,4,0,

0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,3,0,0,

1062,0,0,34,165,0,5,542,0,0,11,11,0,1,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,

0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,4,0,0,1,3,0,0,

1121,0,0,35,0,0,29,550,0,0,0,0,0,1,0,

1136,0,0,0,0,0,27,592,0,0,4,4,0,1,0,1,97,0,0,1,97,0,0,1,10,0,0,1,10,0,0,

1167,0,0,37,66,0,4,692,0,0,3,1,0,1,0,2,3,0,0,2,3,0,0,1,3,0,0,

1194,0,0,38,52,0,4,693,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

1217,0,0,39,36,0,2,694,0,0,1,1,0,1,0,1,3,0,0,

1236,0,0,40,37,0,2,695,0,0,1,1,0,1,0,1,3,0,0,

1255,0,0,41,36,0,2,696,0,0,1,1,0,1,0,1,3,0,0,

1274,0,0,42,37,0,2,697,0,0,1,1,0,1,0,1,3,0,0,

1293,0,0,43,40,0,2,698,0,0,1,1,0,1,0,1,3,0,0,

1312,0,0,44,0,0,29,699,0,0,0,0,0,1,0,

1327,0,0,45,66,0,4,792,0,0,3,1,0,1,0,2,3,0,0,2,3,0,0,1,3,0,0,

1354,0,0,46,52,0,4,793,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

1377,0,0,47,36,0,2,794,0,0,1,1,0,1,0,1,3,0,0,

1396,0,0,48,37,0,2,795,0,0,1,1,0,1,0,1,3,0,0,

1415,0,0,49,36,0,2,796,0,0,1,1,0,1,0,1,3,0,0,

1434,0,0,50,56,0,2,797,0,0,2,2,0,1,0,1,3,0,0,1,97,0,0,

1457,0,0,51,0,0,29,798,0,0,0,0,0,1,0,

1472,0,0,52,85,0,4,915,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,3,0,0,1,97,0,0,

1503,0,0,53,52,0,4,916,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

1526,0,0,54,36,0,2,917,0,0,1,1,0,1,0,1,3,0,0,

1545,0,0,55,37,0,2,918,0,0,1,1,0,1,0,1,3,0,0,

1564,0,0,56,36,0,2,919,0,0,1,1,0,1,0,1,3,0,0,

1583,0,0,57,0,0,29,920,0,0,0,0,0,1,0,

1598,0,0,58,85,0,4,1043,0,0,4,2,0,1,0,2,3,0,0,2,3,0,0,1,3,0,0,1,97,0,0,

1629,0,0,59,52,0,4,1044,0,0,2,1,0,1,0,2,3,0,0,1,3,0,0,

1652,0,0,60,36,0,2,1045,0,0,1,1,0,1,0,1,3,0,0,

1671,0,0,61,0,0,29,1046,0,0,0,0,0,1,0,

};

/\* exec sql begin declare section; \*/

char u[20]="system",p[20]="richu0903";

int acc,pcode;

int temp,cid,und;

char cphone[12];

char name[30],city[20],add[100];

int xacc,xcall,xbill;

int ri;

char xphone[12];

int xsms/\*I\*/, xmms/\*I\*/;

float xsmsr,xmmsr, xgprskb/\*I\*/,xgprsrate,xstdmin/\*I\*/, xnatroamim/\*I\*/, xnatroamom/\*I\*/, xnatroamir, xnatroamor, xstdrate, xisdmin/\*I\*/;

float xinterroami/\*I\*/, xinterroamo/\*I\*/, xinterroamir,xinterroamor;

float xisdrate,xgrandtotal;

long int xaccount;

int xbank;

char xmethod[20];

/\* exec sql end declare section; \*/

/\* exec sql include sqlca;

 \*/

/\*

 \* $Header: sqlca.h 24-apr-2003.12:50:58 mkandarp Exp $ sqlca.h

 \*/

/\* Copyright (c) 1985, 2003, Oracle Corporation.  All rights reserved.  \*/

/\*

NAME

  SQLCA : SQL Communications Area.

FUNCTION

  Contains no code. Oracle fills in the SQLCA with status info

  during the execution of a SQL stmt.

NOTES

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*\*\*                                                        \*\*\*

  \*\*\* This file is SOSD.  Porters must change the data types \*\*\*

  \*\*\* appropriately on their platform.  See notes/pcport.doc \*\*\*

  \*\*\* for more information.                                  \*\*\*

  \*\*\*                                                        \*\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  If the symbol SQLCA\_STORAGE\_CLASS is defined, then the SQLCA

  will be defined to have this storage class. For example:

    #define SQLCA\_STORAGE\_CLASS extern

  will define the SQLCA as an extern.

  If the symbol SQLCA\_INIT is defined, then the SQLCA will be

  statically initialized. Although this is not necessary in order

  to use the SQLCA, it is a good pgming practice not to have

  unitialized variables. However, some C compilers/OS's don't

  allow automatic variables to be init'd in this manner. Therefore,

  if you are INCLUDE'ing the SQLCA in a place where it would be

  an automatic AND your C compiler/OS doesn't allow this style

  of initialization, then SQLCA\_INIT should be left undefined --

  all others can define SQLCA\_INIT if they wish.

  If the symbol SQLCA\_NONE is defined, then the SQLCA variable will

  not be defined at all.  The symbol SQLCA\_NONE should not be defined

  in source modules that have embedded SQL.  However, source modules

  that have no embedded SQL, but need to manipulate a sqlca struct

  passed in as a parameter, can set the SQLCA\_NONE symbol to avoid

  creation of an extraneous sqlca variable.

MODIFIED

    lvbcheng   07/31/98 -  long to int

    jbasu      12/12/94 -  Bug 217878: note this is an SOSD file

    losborne   08/11/92 -  No sqlca var if SQLCA\_NONE macro set

  Clare      12/06/84 - Ch SQLCA to not be an extern.

  Clare      10/21/85 - Add initialization.

  Bradbury   01/05/86 - Only initialize when SQLCA\_INIT set

  Clare      06/12/86 - Add SQLCA\_STORAGE\_CLASS option.

\*/

#ifndef SQLCA

#define SQLCA 1

struct   sqlca

         {

         /\* ub1 \*/ char    sqlcaid[8];

         /\* b4  \*/ int     sqlabc;

         /\* b4  \*/ int     sqlcode;

         struct

           {

           /\* ub2 \*/ unsigned short sqlerrml;

           /\* ub1 \*/ char           sqlerrmc[70];

           } sqlerrm;

         /\* ub1 \*/ char    sqlerrp[8];

         /\* b4  \*/ int     sqlerrd[6];

         /\* ub1 \*/ char    sqlwarn[8];

         /\* ub1 \*/ char    sqlext[8];

         };

#ifndef SQLCA\_NONE

#ifdef   SQLCA\_STORAGE\_CLASS

SQLCA\_STORAGE\_CLASS struct sqlca sqlca

#else

         struct sqlca sqlca

#endif

#ifdef  SQLCA\_INIT

         = {

         {'S', 'Q', 'L', 'C', 'A', ' ', ' ', ' '},

         sizeof(struct sqlca),

         0,

         { 0, {0}},

         {'N', 'O', 'T', ' ', 'S', 'E', 'T', ' '},

         {0, 0, 0, 0, 0, 0},

         {0, 0, 0, 0, 0, 0, 0, 0},

         {0, 0, 0, 0, 0, 0, 0, 0}

         }

#endif

         ;

#endif

#endif

/\* end SQLCA \*/

#include<iostream>

#include<windows.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

#include<cstring>

#include<stdlib.h>

using namespace std;

void gotoxy(int x, int y)

{

  static HANDLE h = NULL;

  if(!h)

    h = GetStdHandle(STD\_OUTPUT\_HANDLE);

  COORD c = { x, y };

  SetConsoleCursorPosition(h,c);

}

void insertbank(char p[10],int a)

{

  for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xphone[10]=NULL;

    xacc=a;

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 4;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (PH\

ONE\_NO=:b2 and ACC\_NO=:b3)";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )5;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xbill;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xcall;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)xphone;

  sqlstm.sqhstl[2] = (unsigned int  )12;

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xacc;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

       cout<<"\t\tEnter account number: ";

       cin>>xaccount;

       cout<<"\n\t\tEnter method of transaction:\n";

       int d;

       cout<<"\t\t1. Credit card \n\t\t2. Debit card\n\t\t3. Net banking\n\t\t4. Cheque\n";

       cout<<"Enter your choice: ";

       cin>>d;

       if(d==1)

       {

            char n[]="Credit card";

            for(int j=0;j<strlen(n);j++)

            xmethod[j]=n[j];

       }

       else if(d==2)

       {

            char n[]="Debit card";

            for(int j=0;j<strlen(n);j++)

            xmethod[j]=n[j];

       }

       else if(d==3)

       {

            char n[]="Net Banking";

            for(int j=0;j<strlen(n);j++)

            xmethod[j]=n[j];

       }

       else if(d==4)

       {

            char n[]="Cheque";

            for(int j=0;j<strlen(n);j++)

            xmethod[j]=n[j];

       }

       else

            cout<<endl<<"Invaild choice ";

        /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 4;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=:b1";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )36;

        sqlstm.selerr = (unsigned short)1;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)&xbank;

        sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)&xcall;

        sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        /\* exec sql insert into bank (BANK\_ID,ACCOUNT,METHOD,ACC\_NO) values (:xbank,:xaccount,:xmethod,:xacc); \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 4;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "insert into bank (BANK\_ID,ACCOUNT,METHOD,ACC\_NO) valu\

es (:b0,:b1,:b2,:b3)";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )59;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)&xbank;

        sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)&xaccount;

        sqlstm.sqhstl[1] = (unsigned int  )sizeof(long);

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqhstv[2] = (         void  \*)xmethod;

        sqlstm.sqhstl[2] = (unsigned int  )20;

        sqlstm.sqhsts[2] = (         int  )0;

        sqlstm.sqindv[2] = (         void  \*)0;

        sqlstm.sqinds[2] = (         int  )0;

        sqlstm.sqharm[2] = (unsigned int  )0;

        sqlstm.sqadto[2] = (unsigned short )0;

        sqlstm.sqtdso[2] = (unsigned short )0;

        sqlstm.sqhstv[3] = (         void  \*)&xacc;

        sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[3] = (         int  )0;

        sqlstm.sqindv[3] = (         void  \*)0;

        sqlstm.sqinds[3] = (         int  )0;

        sqlstm.sqharm[3] = (unsigned int  )0;

        sqlstm.sqadto[3] = (unsigned short )0;

        sqlstm.sqtdso[3] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

}

void selectbank(char p[10],int a)

{

  for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xphone[10]=NULL;

    xacc=a;

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 4;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (PH\

ONE\_NO=:b2 and ACC\_NO=:b3)";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )90;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xbill;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xcall;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)xphone;

  sqlstm.sqhstl[2] = (unsigned int  )12;

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xacc;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 4;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=:b1";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )121;

        sqlstm.selerr = (unsigned short)1;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)&xbank;

        sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)&xcall;

        sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        if(sqlca.sqlcode==0)

        {

        /\* exec sql select ACCOUNT,METHOD,ACC\_NO INTO :xaccount,:xmethod,xacc from bank where BANK\_ID = :xbank; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 4;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "select ACCOUNT ,METHOD ,ACC\_NO into :b0,:b1,:b2  from\

 bank where BANK\_ID=:b3";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )144;

        sqlstm.selerr = (unsigned short)1;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)&xaccount;

        sqlstm.sqhstl[0] = (unsigned int  )sizeof(long);

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)xmethod;

        sqlstm.sqhstl[1] = (unsigned int  )20;

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqhstv[2] = (         void  \*)&xacc;

        sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[2] = (         int  )0;

        sqlstm.sqindv[2] = (         void  \*)0;

        sqlstm.sqinds[2] = (         int  )0;

        sqlstm.sqharm[2] = (unsigned int  )0;

        sqlstm.sqadto[2] = (unsigned short )0;

        sqlstm.sqtdso[2] = (unsigned short )0;

        sqlstm.sqhstv[3] = (         void  \*)&xbank;

        sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[3] = (         int  )0;

        sqlstm.sqindv[3] = (         void  \*)0;

        sqlstm.sqinds[3] = (         int  )0;

        sqlstm.sqharm[3] = (unsigned int  )0;

        sqlstm.sqadto[3] = (unsigned short )0;

        sqlstm.sqtdso[3] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        /\* exec sql select cname,caddr,ccity,cpcode into :name,:add,:city,:pcode from Tcustomer where acc\_no=:acc; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 5;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "select cname ,caddr ,ccity ,cpcode into :b0,:b1,:b2,:\

b3  from Tcustomer where acc\_no=:b4";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )175;

        sqlstm.selerr = (unsigned short)1;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)name;

        sqlstm.sqhstl[0] = (unsigned int  )30;

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)add;

        sqlstm.sqhstl[1] = (unsigned int  )100;

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqhstv[2] = (         void  \*)city;

        sqlstm.sqhstl[2] = (unsigned int  )20;

        sqlstm.sqhsts[2] = (         int  )0;

        sqlstm.sqindv[2] = (         void  \*)0;

        sqlstm.sqinds[2] = (         int  )0;

        sqlstm.sqharm[2] = (unsigned int  )0;

        sqlstm.sqadto[2] = (unsigned short )0;

        sqlstm.sqtdso[2] = (unsigned short )0;

        sqlstm.sqhstv[3] = (         void  \*)&pcode;

        sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[3] = (         int  )0;

        sqlstm.sqindv[3] = (         void  \*)0;

        sqlstm.sqinds[3] = (         int  )0;

        sqlstm.sqharm[3] = (unsigned int  )0;

        sqlstm.sqadto[3] = (unsigned short )0;

        sqlstm.sqtdso[3] = (unsigned short )0;

        sqlstm.sqhstv[4] = (         void  \*)&acc;

        sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[4] = (         int  )0;

        sqlstm.sqindv[4] = (         void  \*)0;

        sqlstm.sqinds[4] = (         int  )0;

        sqlstm.sqharm[4] = (unsigned int  )0;

        sqlstm.sqadto[4] = (unsigned short )0;

        sqlstm.sqtdso[4] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        cout<<"\n\t\tCustomer Id : "<<xacc<<endl<<"\t\t\tName = "<<name<<endl<<"\t\t\tCity = "<<city<<endl;

        cout<<"\t\t\tAddress = "<<add<<endl<<"\t\t\tPin code = "<<pcode<<endl;

        cout<<"\t\tBank id : "<<xbank<<endl<<"\t\tAccount : "<<xaccount<<endl<<"\t\tMethod : "<<xmethod<<endl;

        }

        else

            cout<<"\n\n\t\t\t\tFILE Not Found.....";

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

}

void editbank(char p[10],int a)

{

    int ch;

    for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xphone[10]=NULL;

    xacc=a;

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 5;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (PH\

ONE\_NO=:b2 and ACC\_NO=:b3)";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )210;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xbill;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xcall;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)xphone;

  sqlstm.sqhstl[2] = (unsigned int  )12;

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xacc;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        cout<<"=============================================================================";

        cout<<"\n\t\t\t Current Customer Bank Details"<<endl;

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

        selectbank(p,a);

        cout<<"=============================================================================";

        cout<<"\n\t\t\t\t  Edit What ??????    "<<endl;

        cout<<endl<<"\t1.Account  2.Method"<<endl;

        cout<<"=============================================================================";

        pp :

        cout<<endl<<"\t\t\t\tEnter Option   :  ";

        cin>>ch;

        if(ch==1)

        {

            cout<<"\n\t\t\t\tEnter Account Number  : ";

            cin>>xaccount;

        }

        if(ch==2)

        {

            cout<<"Enter method of transaction:\n";

            int d;

            cout<<"\t\t1. Credit card \n\t\t2. Debit card\n\t\t3. Net banking\n\t\t4. Cheque\n";

            cout<<"enter your choice";

            cin>>d;

            if(d==1)

            {

                char n[]="Credit card";

                for(int j=0;j<strlen(n);j++)

                xmethod[j]=n[j];

            }

            else if(d==2)

            {

                char n[]="Debit card";

                for(int j=0;j<strlen(n);j++)

                xmethod[j]=n[j];

            }

            else if(d==3)

            {

                char n[]="Net Banking";

                for(int j=0;j<strlen(n);j++)

                xmethod[j]=n[j];

            }

            else if(d==4)

            {

                char n[]="Cheque";

                for(int j=0;j<strlen(n);j++)

                xmethod[j]=n[j];

            }

            else

                cout<<endl<<"Invaild choice ";

        }

        /\* exec sql update bank set ACCOUNT = :xaccount, method = :xmethod where bank\_id=:xbank and ACC\_NO=:xacc; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 5;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.stmt = "update bank  set ACCOUNT=:b0,method=:b1 where (bank\_i\

d=:b2 and ACC\_NO=:b3)";

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )241;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlstm.sqhstv[0] = (         void  \*)&xaccount;

        sqlstm.sqhstl[0] = (unsigned int  )sizeof(long);

        sqlstm.sqhsts[0] = (         int  )0;

        sqlstm.sqindv[0] = (         void  \*)0;

        sqlstm.sqinds[0] = (         int  )0;

        sqlstm.sqharm[0] = (unsigned int  )0;

        sqlstm.sqadto[0] = (unsigned short )0;

        sqlstm.sqtdso[0] = (unsigned short )0;

        sqlstm.sqhstv[1] = (         void  \*)xmethod;

        sqlstm.sqhstl[1] = (unsigned int  )20;

        sqlstm.sqhsts[1] = (         int  )0;

        sqlstm.sqindv[1] = (         void  \*)0;

        sqlstm.sqinds[1] = (         int  )0;

        sqlstm.sqharm[1] = (unsigned int  )0;

        sqlstm.sqadto[1] = (unsigned short )0;

        sqlstm.sqtdso[1] = (unsigned short )0;

        sqlstm.sqhstv[2] = (         void  \*)&xbank;

        sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[2] = (         int  )0;

        sqlstm.sqindv[2] = (         void  \*)0;

        sqlstm.sqinds[2] = (         int  )0;

        sqlstm.sqharm[2] = (unsigned int  )0;

        sqlstm.sqadto[2] = (unsigned short )0;

        sqlstm.sqtdso[2] = (unsigned short )0;

        sqlstm.sqhstv[3] = (         void  \*)&xacc;

        sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

        sqlstm.sqhsts[3] = (         int  )0;

        sqlstm.sqindv[3] = (         void  \*)0;

        sqlstm.sqinds[3] = (         int  )0;

        sqlstm.sqharm[3] = (unsigned int  )0;

        sqlstm.sqadto[3] = (unsigned short )0;

        sqlstm.sqtdso[3] = (unsigned short )0;

        sqlstm.sqphsv = sqlstm.sqhstv;

        sqlstm.sqphsl = sqlstm.sqhstl;

        sqlstm.sqphss = sqlstm.sqhsts;

        sqlstm.sqpind = sqlstm.sqindv;

        sqlstm.sqpins = sqlstm.sqinds;

        sqlstm.sqparm = sqlstm.sqharm;

        sqlstm.sqparc = sqlstm.sqharc;

        sqlstm.sqpadto = sqlstm.sqadto;

        sqlstm.sqptdso = sqlstm.sqtdso;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

                if(sqlca.sqlcode==0)

    {

        cout<<"Success"<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

        /\* exec sql commit work; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 5;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )272;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    }

}

void insertphone(int a)

{

    int count=0;

    xacc=a;

    j:

    cout<<"\nEnter phone: "<<endl;

    cin>>xphone;

    for(int i = 0; i < strlen(xphone); i++)

        if(xphone[i] != '\0')

            count++;

    xphone[11]=NULL;

    if(count==10)

    {

        /\* exec sql select cname,caddr,ccity,cpcode into :name,:add,:city,:pcode from Tcustomer where acc\_no=:xacc; \*/

{

      struct sqlexd sqlstm;

      sqlstm.sqlvsn = 12;

      sqlstm.arrsiz = 5;

      sqlstm.sqladtp = &sqladt;

      sqlstm.sqltdsp = &sqltds;

      sqlstm.stmt = "select cname ,caddr ,ccity ,cpcode into :b0,:b1,:b2,:b3\

  from Tcustomer where acc\_no=:b4";

      sqlstm.iters = (unsigned int  )1;

      sqlstm.offset = (unsigned int  )287;

      sqlstm.selerr = (unsigned short)1;

      sqlstm.cud = sqlcud0;

      sqlstm.sqlest = (unsigned char  \*)&sqlca;

      sqlstm.sqlety = (unsigned short)4352;

      sqlstm.occurs = (unsigned int  )0;

      sqlstm.sqhstv[0] = (         void  \*)name;

      sqlstm.sqhstl[0] = (unsigned int  )30;

      sqlstm.sqhsts[0] = (         int  )0;

      sqlstm.sqindv[0] = (         void  \*)0;

      sqlstm.sqinds[0] = (         int  )0;

      sqlstm.sqharm[0] = (unsigned int  )0;

      sqlstm.sqadto[0] = (unsigned short )0;

      sqlstm.sqtdso[0] = (unsigned short )0;

      sqlstm.sqhstv[1] = (         void  \*)add;

      sqlstm.sqhstl[1] = (unsigned int  )100;

      sqlstm.sqhsts[1] = (         int  )0;

      sqlstm.sqindv[1] = (         void  \*)0;

      sqlstm.sqinds[1] = (         int  )0;

      sqlstm.sqharm[1] = (unsigned int  )0;

      sqlstm.sqadto[1] = (unsigned short )0;

      sqlstm.sqtdso[1] = (unsigned short )0;

      sqlstm.sqhstv[2] = (         void  \*)city;

      sqlstm.sqhstl[2] = (unsigned int  )20;

      sqlstm.sqhsts[2] = (         int  )0;

      sqlstm.sqindv[2] = (         void  \*)0;

      sqlstm.sqinds[2] = (         int  )0;

      sqlstm.sqharm[2] = (unsigned int  )0;

      sqlstm.sqadto[2] = (unsigned short )0;

      sqlstm.sqtdso[2] = (unsigned short )0;

      sqlstm.sqhstv[3] = (         void  \*)&pcode;

      sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

      sqlstm.sqhsts[3] = (         int  )0;

      sqlstm.sqindv[3] = (         void  \*)0;

      sqlstm.sqinds[3] = (         int  )0;

      sqlstm.sqharm[3] = (unsigned int  )0;

      sqlstm.sqadto[3] = (unsigned short )0;

      sqlstm.sqtdso[3] = (unsigned short )0;

      sqlstm.sqhstv[4] = (         void  \*)&xacc;

      sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

      sqlstm.sqhsts[4] = (         int  )0;

      sqlstm.sqindv[4] = (         void  \*)0;

      sqlstm.sqinds[4] = (         int  )0;

      sqlstm.sqharm[4] = (unsigned int  )0;

      sqlstm.sqadto[4] = (unsigned short )0;

      sqlstm.sqtdso[4] = (unsigned short )0;

      sqlstm.sqphsv = sqlstm.sqhstv;

      sqlstm.sqphsl = sqlstm.sqhstl;

      sqlstm.sqphss = sqlstm.sqhsts;

      sqlstm.sqpind = sqlstm.sqindv;

      sqlstm.sqpins = sqlstm.sqinds;

      sqlstm.sqparm = sqlstm.sqharm;

      sqlstm.sqparc = sqlstm.sqharc;

      sqlstm.sqpadto = sqlstm.sqadto;

      sqlstm.sqptdso = sqlstm.sqtdso;

      sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        if(sqlca.sqlcode==0)

        {

            cout<<"\t\t\tName = "<<name<<endl<<"\t\t\tPhone Number = "<<xphone<<endl<<"\t\t\t Account number = "<<xacc<<endl;

            /\* exec sql select seqbill.nextval into :xbill from dual; \*/

{

       struct sqlexd sqlstm;

       sqlstm.sqlvsn = 12;

       sqlstm.arrsiz = 5;

       sqlstm.sqladtp = &sqladt;

       sqlstm.sqltdsp = &sqltds;

       sqlstm.stmt = "select seqbill.nextval  into :b0  from dual ";

       sqlstm.iters = (unsigned int  )1;

       sqlstm.offset = (unsigned int  )322;

       sqlstm.selerr = (unsigned short)1;

       sqlstm.cud = sqlcud0;

       sqlstm.sqlest = (unsigned char  \*)&sqlca;

       sqlstm.sqlety = (unsigned short)4352;

       sqlstm.occurs = (unsigned int  )0;

       sqlstm.sqhstv[0] = (         void  \*)&xbill;

       sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

       sqlstm.sqhsts[0] = (         int  )0;

       sqlstm.sqindv[0] = (         void  \*)0;

       sqlstm.sqinds[0] = (         int  )0;

       sqlstm.sqharm[0] = (unsigned int  )0;

       sqlstm.sqadto[0] = (unsigned short )0;

       sqlstm.sqtdso[0] = (unsigned short )0;

       sqlstm.sqphsv = sqlstm.sqhstv;

       sqlstm.sqphsl = sqlstm.sqhstl;

       sqlstm.sqphss = sqlstm.sqhsts;

       sqlstm.sqpind = sqlstm.sqindv;

       sqlstm.sqpins = sqlstm.sqinds;

       sqlstm.sqparm = sqlstm.sqharm;

       sqlstm.sqparc = sqlstm.sqharc;

       sqlstm.sqpadto = sqlstm.sqadto;

       sqlstm.sqptdso = sqlstm.sqtdso;

       sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql select seqcall.nextval into :xcall from dual; \*/

{

       struct sqlexd sqlstm;

       sqlstm.sqlvsn = 12;

       sqlstm.arrsiz = 5;

       sqlstm.sqladtp = &sqladt;

       sqlstm.sqltdsp = &sqltds;

       sqlstm.stmt = "select seqcall.nextval  into :b0  from dual ";

       sqlstm.iters = (unsigned int  )1;

       sqlstm.offset = (unsigned int  )341;

       sqlstm.selerr = (unsigned short)1;

       sqlstm.cud = sqlcud0;

       sqlstm.sqlest = (unsigned char  \*)&sqlca;

       sqlstm.sqlety = (unsigned short)4352;

       sqlstm.occurs = (unsigned int  )0;

       sqlstm.sqhstv[0] = (         void  \*)&xcall;

       sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

       sqlstm.sqhsts[0] = (         int  )0;

       sqlstm.sqindv[0] = (         void  \*)0;

       sqlstm.sqinds[0] = (         int  )0;

       sqlstm.sqharm[0] = (unsigned int  )0;

       sqlstm.sqadto[0] = (unsigned short )0;

       sqlstm.sqtdso[0] = (unsigned short )0;

       sqlstm.sqphsv = sqlstm.sqhstv;

       sqlstm.sqphsl = sqlstm.sqhstl;

       sqlstm.sqphss = sqlstm.sqhsts;

       sqlstm.sqpind = sqlstm.sqindv;

       sqlstm.sqpins = sqlstm.sqinds;

       sqlstm.sqparm = sqlstm.sqharm;

       sqlstm.sqparc = sqlstm.sqharc;

       sqlstm.sqpadto = sqlstm.sqadto;

       sqlstm.sqptdso = sqlstm.sqtdso;

       sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql insert into tphone (PHONE\_NO,ACC\_NO,BILL\_ID,CALL\_ID) values (:xphone,:xacc,:xbill,:xcall); \*/

{

       struct sqlexd sqlstm;

       sqlstm.sqlvsn = 12;

       sqlstm.arrsiz = 5;

       sqlstm.sqladtp = &sqladt;

       sqlstm.sqltdsp = &sqltds;

       sqlstm.stmt = "insert into tphone (PHONE\_NO,ACC\_NO,BILL\_ID,CALL\_ID) v\

alues (:b0,:b1,:b2,:b3)";

       sqlstm.iters = (unsigned int  )1;

       sqlstm.offset = (unsigned int  )360;

       sqlstm.cud = sqlcud0;

       sqlstm.sqlest = (unsigned char  \*)&sqlca;

       sqlstm.sqlety = (unsigned short)4352;

       sqlstm.occurs = (unsigned int  )0;

       sqlstm.sqhstv[0] = (         void  \*)xphone;

       sqlstm.sqhstl[0] = (unsigned int  )12;

       sqlstm.sqhsts[0] = (         int  )0;

       sqlstm.sqindv[0] = (         void  \*)0;

       sqlstm.sqinds[0] = (         int  )0;

       sqlstm.sqharm[0] = (unsigned int  )0;

       sqlstm.sqadto[0] = (unsigned short )0;

       sqlstm.sqtdso[0] = (unsigned short )0;

       sqlstm.sqhstv[1] = (         void  \*)&xacc;

       sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

       sqlstm.sqhsts[1] = (         int  )0;

       sqlstm.sqindv[1] = (         void  \*)0;

       sqlstm.sqinds[1] = (         int  )0;

       sqlstm.sqharm[1] = (unsigned int  )0;

       sqlstm.sqadto[1] = (unsigned short )0;

       sqlstm.sqtdso[1] = (unsigned short )0;

       sqlstm.sqhstv[2] = (         void  \*)&xbill;

       sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

       sqlstm.sqhsts[2] = (         int  )0;

       sqlstm.sqindv[2] = (         void  \*)0;

       sqlstm.sqinds[2] = (         int  )0;

       sqlstm.sqharm[2] = (unsigned int  )0;

       sqlstm.sqadto[2] = (unsigned short )0;

       sqlstm.sqtdso[2] = (unsigned short )0;

       sqlstm.sqhstv[3] = (         void  \*)&xcall;

       sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

       sqlstm.sqhsts[3] = (         int  )0;

       sqlstm.sqindv[3] = (         void  \*)0;

       sqlstm.sqinds[3] = (         int  )0;

       sqlstm.sqharm[3] = (unsigned int  )0;

       sqlstm.sqadto[3] = (unsigned short )0;

       sqlstm.sqtdso[3] = (unsigned short )0;

       sqlstm.sqphsv = sqlstm.sqhstv;

       sqlstm.sqphsl = sqlstm.sqhstl;

       sqlstm.sqphss = sqlstm.sqhsts;

       sqlstm.sqpind = sqlstm.sqindv;

       sqlstm.sqpins = sqlstm.sqinds;

       sqlstm.sqparm = sqlstm.sqharm;

       sqlstm.sqparc = sqlstm.sqharc;

       sqlstm.sqpadto = sqlstm.sqadto;

       sqlstm.sqptdso = sqlstm.sqtdso;

       sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            if(sqlca.sqlcode==0)

            {

                cout<<"success";

            }

            else

                cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl;

            /\* exec sql commit work ; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 5;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )391;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        }

        else

            cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

    }

    else

    {

        cout<<"\nInvalid input for phone number try again";

        goto j;

    }

}

void selectphone(int a)

{

    xacc=a;

    /\* exec sql declare t1 cursor for select phone\_no from tphone where acc\_no=:xacc; \*/

    int cou=0;

    /\* exec sql open t1; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = sq0016;

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )406;

    sqlstm.selerr = (unsigned short)1;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqcmod = (unsigned int )0;

    sqlstm.sqhstv[0] = (         void  \*)&xacc;

    sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    do

    {

        if (sqlca.sqlcode == 1403) break;

        else

        {

            cou++;

            /\* exec sql fetch t1 into :xphone; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 5;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )425;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqfoff = (           int )0;

            sqlstm.sqfmod = (unsigned int )2;

            sqlstm.sqhstv[0] = (         void  \*)xphone;

            sqlstm.sqhstl[0] = (unsigned int  )12;

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            cout<<"\t\tPhone number "<<cou<<" is= "<<xphone<<endl;

        }

    }while(1);

    /\* exec sql close t1; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )444;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

}

void selectcust()

{

    cout<<"\nEnter account number:";

    cin>>acc;

    /\* exec sql select cname,caddr,ccity,cpcode into :name,:add,:city,:pcode from Tcustomer where acc\_no=:acc; \*/

{

 struct sqlexd sqlstm;

 sqlstm.sqlvsn = 12;

 sqlstm.arrsiz = 5;

 sqlstm.sqladtp = &sqladt;

 sqlstm.sqltdsp = &sqltds;

 sqlstm.stmt = "select cname ,caddr ,ccity ,cpcode into :b0,:b1,:b2,:b3  fro\

m Tcustomer where acc\_no=:b4";

 sqlstm.iters = (unsigned int  )1;

 sqlstm.offset = (unsigned int  )459;

 sqlstm.selerr = (unsigned short)1;

 sqlstm.cud = sqlcud0;

 sqlstm.sqlest = (unsigned char  \*)&sqlca;

 sqlstm.sqlety = (unsigned short)4352;

 sqlstm.occurs = (unsigned int  )0;

 sqlstm.sqhstv[0] = (         void  \*)name;

 sqlstm.sqhstl[0] = (unsigned int  )30;

 sqlstm.sqhsts[0] = (         int  )0;

 sqlstm.sqindv[0] = (         void  \*)0;

 sqlstm.sqinds[0] = (         int  )0;

 sqlstm.sqharm[0] = (unsigned int  )0;

 sqlstm.sqadto[0] = (unsigned short )0;

 sqlstm.sqtdso[0] = (unsigned short )0;

 sqlstm.sqhstv[1] = (         void  \*)add;

 sqlstm.sqhstl[1] = (unsigned int  )100;

 sqlstm.sqhsts[1] = (         int  )0;

 sqlstm.sqindv[1] = (         void  \*)0;

 sqlstm.sqinds[1] = (         int  )0;

 sqlstm.sqharm[1] = (unsigned int  )0;

 sqlstm.sqadto[1] = (unsigned short )0;

 sqlstm.sqtdso[1] = (unsigned short )0;

 sqlstm.sqhstv[2] = (         void  \*)city;

 sqlstm.sqhstl[2] = (unsigned int  )20;

 sqlstm.sqhsts[2] = (         int  )0;

 sqlstm.sqindv[2] = (         void  \*)0;

 sqlstm.sqinds[2] = (         int  )0;

 sqlstm.sqharm[2] = (unsigned int  )0;

 sqlstm.sqadto[2] = (unsigned short )0;

 sqlstm.sqtdso[2] = (unsigned short )0;

 sqlstm.sqhstv[3] = (         void  \*)&pcode;

 sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

 sqlstm.sqhsts[3] = (         int  )0;

 sqlstm.sqindv[3] = (         void  \*)0;

 sqlstm.sqinds[3] = (         int  )0;

 sqlstm.sqharm[3] = (unsigned int  )0;

 sqlstm.sqadto[3] = (unsigned short )0;

 sqlstm.sqtdso[3] = (unsigned short )0;

 sqlstm.sqhstv[4] = (         void  \*)&acc;

 sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

 sqlstm.sqhsts[4] = (         int  )0;

 sqlstm.sqindv[4] = (         void  \*)0;

 sqlstm.sqinds[4] = (         int  )0;

 sqlstm.sqharm[4] = (unsigned int  )0;

 sqlstm.sqadto[4] = (unsigned short )0;

 sqlstm.sqtdso[4] = (unsigned short )0;

 sqlstm.sqphsv = sqlstm.sqhstv;

 sqlstm.sqphsl = sqlstm.sqhstl;

 sqlstm.sqphss = sqlstm.sqhsts;

 sqlstm.sqpind = sqlstm.sqindv;

 sqlstm.sqpins = sqlstm.sqinds;

 sqlstm.sqparm = sqlstm.sqharm;

 sqlstm.sqparc = sqlstm.sqharc;

 sqlstm.sqpadto = sqlstm.sqadto;

 sqlstm.sqptdso = sqlstm.sqtdso;

 sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        cout<<"\t\t\tName = "<<name<<endl<<"\t\t\tCity = "<<city<<endl;

        cout<<"\t\t\tAddress = "<<add<<endl<<"\t\t\tPin code = "<<pcode<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

}

void insertcust()

{

    cout<<"Enter name: ";

    cin>>name;

    cout<<"Enter city: ";

    cin>>city;

    cout<<"Enter address: ";

    cin>>add;

    cout<<"Enter pincode: ";

    cin>>pcode;

    /\* exec sql select seqacc.nextval into :acc from dual; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 5;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select seqacc.nextval  into :b0  from dual ";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )494;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&acc;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql insert into Tcustomer(acc\_no,cname,caddr,ccity,cpcode) values(:acc,:name,:add,:city,:pcode); \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = "insert into Tcustomer (acc\_no,cname,caddr,ccity,cpcode) v\

alues (:b0,:b1,:b2,:b3,:b4)";

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )513;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqhstv[0] = (         void  \*)&acc;

    sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqhstv[1] = (         void  \*)name;

    sqlstm.sqhstl[1] = (unsigned int  )30;

    sqlstm.sqhsts[1] = (         int  )0;

    sqlstm.sqindv[1] = (         void  \*)0;

    sqlstm.sqinds[1] = (         int  )0;

    sqlstm.sqharm[1] = (unsigned int  )0;

    sqlstm.sqadto[1] = (unsigned short )0;

    sqlstm.sqtdso[1] = (unsigned short )0;

    sqlstm.sqhstv[2] = (         void  \*)add;

    sqlstm.sqhstl[2] = (unsigned int  )100;

    sqlstm.sqhsts[2] = (         int  )0;

    sqlstm.sqindv[2] = (         void  \*)0;

    sqlstm.sqinds[2] = (         int  )0;

    sqlstm.sqharm[2] = (unsigned int  )0;

    sqlstm.sqadto[2] = (unsigned short )0;

    sqlstm.sqtdso[2] = (unsigned short )0;

    sqlstm.sqhstv[3] = (         void  \*)city;

    sqlstm.sqhstl[3] = (unsigned int  )20;

    sqlstm.sqhsts[3] = (         int  )0;

    sqlstm.sqindv[3] = (         void  \*)0;

    sqlstm.sqinds[3] = (         int  )0;

    sqlstm.sqharm[3] = (unsigned int  )0;

    sqlstm.sqadto[3] = (unsigned short )0;

    sqlstm.sqtdso[3] = (unsigned short )0;

    sqlstm.sqhstv[4] = (         void  \*)&pcode;

    sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[4] = (         int  )0;

    sqlstm.sqindv[4] = (         void  \*)0;

    sqlstm.sqinds[4] = (         int  )0;

    sqlstm.sqharm[4] = (unsigned int  )0;

    sqlstm.sqadto[4] = (unsigned short )0;

    sqlstm.sqtdso[4] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        if(sqlca.sqlcode==0)

    {

        cout<<"Success"<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

    cout<<"Your account number is :"<<acc<<endl;

    /\* exec sql commit work ; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )548;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

}

void editcust(int a)

{

    system("cls");

    int ch;

    acc=a;

    cout<<"=============================================================================";

    cout<<"\n\t\t\t Current Customer Details"<<endl;

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

    /\* exec sql select cname,caddr,ccity,cpcode into :name,:add,:city,:pcode from Tcustomer where acc\_no=:acc; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = "select cname ,caddr ,ccity ,cpcode into :b0,:b1,:b2,:b3  \

from Tcustomer where acc\_no=:b4";

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )563;

    sqlstm.selerr = (unsigned short)1;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqhstv[0] = (         void  \*)name;

    sqlstm.sqhstl[0] = (unsigned int  )30;

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqhstv[1] = (         void  \*)add;

    sqlstm.sqhstl[1] = (unsigned int  )100;

    sqlstm.sqhsts[1] = (         int  )0;

    sqlstm.sqindv[1] = (         void  \*)0;

    sqlstm.sqinds[1] = (         int  )0;

    sqlstm.sqharm[1] = (unsigned int  )0;

    sqlstm.sqadto[1] = (unsigned short )0;

    sqlstm.sqtdso[1] = (unsigned short )0;

    sqlstm.sqhstv[2] = (         void  \*)city;

    sqlstm.sqhstl[2] = (unsigned int  )20;

    sqlstm.sqhsts[2] = (         int  )0;

    sqlstm.sqindv[2] = (         void  \*)0;

    sqlstm.sqinds[2] = (         int  )0;

    sqlstm.sqharm[2] = (unsigned int  )0;

    sqlstm.sqadto[2] = (unsigned short )0;

    sqlstm.sqtdso[2] = (unsigned short )0;

    sqlstm.sqhstv[3] = (         void  \*)&pcode;

    sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[3] = (         int  )0;

    sqlstm.sqindv[3] = (         void  \*)0;

    sqlstm.sqinds[3] = (         int  )0;

    sqlstm.sqharm[3] = (unsigned int  )0;

    sqlstm.sqadto[3] = (unsigned short )0;

    sqlstm.sqtdso[3] = (unsigned short )0;

    sqlstm.sqhstv[4] = (         void  \*)&acc;

    sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[4] = (         int  )0;

    sqlstm.sqindv[4] = (         void  \*)0;

    sqlstm.sqinds[4] = (         int  )0;

    sqlstm.sqharm[4] = (unsigned int  )0;

    sqlstm.sqadto[4] = (unsigned short )0;

    sqlstm.sqtdso[4] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        cout<<"\n\t\t\tName = "<<name<<endl<<"\t\t\tAddress = "<<add<<endl;

        cout<<"\t\t\tCity = "<<city<<endl<<"\t\t\tPin code = "<<pcode<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

    cout<<"=============================================================================";

    cout<<"\n\t\t\t\t  Edit What ??????    "<<endl;

    cout<<endl<<"\t1.Name  2.Address  3.City  4.Pin Code "<<endl;

    cout<<"=============================================================================";

    pp :

    cout<<endl<<"\t\t\t\tEnter Option   :  ";

    cin>>ch;

    if(ch==1)

    {

        cout<<"\n\t\t\t\tEnter Name  : ";

        cin>>name;

    }

    if(ch==2)

    {

        cout<<"\n\t\t\t\tEnter Address : ";

        cin>>add;

    }

    if(ch==3)

    {

        cout<<"\n\t\t\t\tEnter City : ";

        cin>>city;

    }

    if(ch==4)

    {

        cout<<"\n\t\t\t\tEnter Pincode : ";

        cin>>pcode;

    }

    /\* exec sql update tcustomer set cname = :name, caddr = :add, ccity = :city, cpcode = :pcode where acc\_no=:acc; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = "update tcustomer  set cname=:b0,caddr=:b1,ccity=:b2,cpcod\

e=:b3 where acc\_no=:b4";

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )598;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqhstv[0] = (         void  \*)name;

    sqlstm.sqhstl[0] = (unsigned int  )30;

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqhstv[1] = (         void  \*)add;

    sqlstm.sqhstl[1] = (unsigned int  )100;

    sqlstm.sqhsts[1] = (         int  )0;

    sqlstm.sqindv[1] = (         void  \*)0;

    sqlstm.sqinds[1] = (         int  )0;

    sqlstm.sqharm[1] = (unsigned int  )0;

    sqlstm.sqadto[1] = (unsigned short )0;

    sqlstm.sqtdso[1] = (unsigned short )0;

    sqlstm.sqhstv[2] = (         void  \*)city;

    sqlstm.sqhstl[2] = (unsigned int  )20;

    sqlstm.sqhsts[2] = (         int  )0;

    sqlstm.sqindv[2] = (         void  \*)0;

    sqlstm.sqinds[2] = (         int  )0;

    sqlstm.sqharm[2] = (unsigned int  )0;

    sqlstm.sqadto[2] = (unsigned short )0;

    sqlstm.sqtdso[2] = (unsigned short )0;

    sqlstm.sqhstv[3] = (         void  \*)&pcode;

    sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[3] = (         int  )0;

    sqlstm.sqindv[3] = (         void  \*)0;

    sqlstm.sqinds[3] = (         int  )0;

    sqlstm.sqharm[3] = (unsigned int  )0;

    sqlstm.sqadto[3] = (unsigned short )0;

    sqlstm.sqtdso[3] = (unsigned short )0;

    sqlstm.sqhstv[4] = (         void  \*)&acc;

    sqlstm.sqhstl[4] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[4] = (         int  )0;

    sqlstm.sqindv[4] = (         void  \*)0;

    sqlstm.sqinds[4] = (         int  )0;

    sqlstm.sqharm[4] = (unsigned int  )0;

    sqlstm.sqadto[4] = (unsigned short )0;

    sqlstm.sqtdso[4] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            if(sqlca.sqlcode==0)

    {

        cout<<"Success"<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

    /\* exec sql commit work; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )633;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

}

void cal()

{

    xstdrate= 1\*xstdmin;

    xnatroamir=2\*xnatroamim;

    xnatroamor=2.5\*xnatroamom;

    xsmsr=0.50\*xsms;

    xmmsr=3\*xmms;

    xgprsrate=0.10\*xgprskb;

    xisdrate= 8\*xisdmin;

    xinterroamir=xinterroami\*17;

    xinterroamor=xinterroamo\*15;

    xgrandtotal=xstdrate + xnatroamir + xnatroamor + xinterroamir + xinterroamor + xisdrate + xgprsrate + xsmsr + xmmsr;

}

void insertbill(char p[10],int a)

{

    for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xphone[10]=NULL;

    xacc=a;

    /\* exec sql select seqbank.nextval into :xbank from dual; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 5;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = "select seqbank.nextval  into :b0  from dual ";

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )648;

    sqlstm.selerr = (unsigned short)1;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqhstv[0] = (         void  \*)&xbank;

    sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 5;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (PH\

ONE\_NO=:b2 and ACC\_NO=:b3)";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )667;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xbill;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xcall;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)xphone;

  sqlstm.sqhstl[2] = (unsigned int  )12;

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xacc;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        cout<<endl<<"  Incoming Calls (mts)        : ";

        cin>>xstdmin;

        cout<<endl<<"  No of SMS                   : ";

        cin>>xsms;

        cout<<endl<<"  Roaming Incoming Calls (mts): ";

        cin>>xnatroamim;

        cout<<endl<<"  Roaming Outgoing Calls (mts): ";

        cin>>xnatroamom;

        cout<<endl<<"  GPRS Usage ( in Kb )        : ";

        cin>>xgprskb;

        cout<<endl<<"  No of MMS                   : ";

        cin>>xmms;

        cout<<endl<<"  ISD Calls (mts)             : ";

        cin>>xisdmin;

        cout<<endl<<"  Inter Roaming Incoming (mts): ";

        cin>>xinterroami;

        cout<<endl<<"  Inter Roaming Outgoing (mts): ";

        cin>>xinterroamo;

        cal();

        /\* exec sql insert into tbill(SMS,MMS,GPRSKP,STDMIN,NATROAMIM,NATROAMOM,ISDMIN,INTERROAMI,INTERROAMO,BILL\_ID) values (:xsms,:xmms,:xgprskb,:xstdmin,:xnatroamim,:xnatroamom,:xisdmin,:xinterroami,:xinterroamo,:xbill); \*/

{

   struct sqlexd sqlstm;

   sqlstm.sqlvsn = 12;

   sqlstm.arrsiz = 10;

   sqlstm.sqladtp = &sqladt;

   sqlstm.sqltdsp = &sqltds;

   sqlstm.stmt = "insert into tbill (SMS,MMS,GPRSKP,STDMIN,NATROAMIM,NATROAM\

OM,ISDMIN,INTERROAMI,INTERROAMO,BILL\_ID) values (:b0,:b1,:b2,:b3,:b4,:b5,:b6,:\

b7,:b8,:b9)";

   sqlstm.iters = (unsigned int  )1;

   sqlstm.offset = (unsigned int  )698;

   sqlstm.cud = sqlcud0;

   sqlstm.sqlest = (unsigned char  \*)&sqlca;

   sqlstm.sqlety = (unsigned short)4352;

   sqlstm.occurs = (unsigned int  )0;

   sqlstm.sqhstv[0] = (         void  \*)&xsms;

   sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

   sqlstm.sqhsts[0] = (         int  )0;

   sqlstm.sqindv[0] = (         void  \*)0;

   sqlstm.sqinds[0] = (         int  )0;

   sqlstm.sqharm[0] = (unsigned int  )0;

   sqlstm.sqadto[0] = (unsigned short )0;

   sqlstm.sqtdso[0] = (unsigned short )0;

   sqlstm.sqhstv[1] = (         void  \*)&xmms;

   sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

   sqlstm.sqhsts[1] = (         int  )0;

   sqlstm.sqindv[1] = (         void  \*)0;

   sqlstm.sqinds[1] = (         int  )0;

   sqlstm.sqharm[1] = (unsigned int  )0;

   sqlstm.sqadto[1] = (unsigned short )0;

   sqlstm.sqtdso[1] = (unsigned short )0;

   sqlstm.sqhstv[2] = (         void  \*)&xgprskb;

   sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[2] = (         int  )0;

   sqlstm.sqindv[2] = (         void  \*)0;

   sqlstm.sqinds[2] = (         int  )0;

   sqlstm.sqharm[2] = (unsigned int  )0;

   sqlstm.sqadto[2] = (unsigned short )0;

   sqlstm.sqtdso[2] = (unsigned short )0;

   sqlstm.sqhstv[3] = (         void  \*)&xstdmin;

   sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[3] = (         int  )0;

   sqlstm.sqindv[3] = (         void  \*)0;

   sqlstm.sqinds[3] = (         int  )0;

   sqlstm.sqharm[3] = (unsigned int  )0;

   sqlstm.sqadto[3] = (unsigned short )0;

   sqlstm.sqtdso[3] = (unsigned short )0;

   sqlstm.sqhstv[4] = (         void  \*)&xnatroamim;

   sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[4] = (         int  )0;

   sqlstm.sqindv[4] = (         void  \*)0;

   sqlstm.sqinds[4] = (         int  )0;

   sqlstm.sqharm[4] = (unsigned int  )0;

   sqlstm.sqadto[4] = (unsigned short )0;

   sqlstm.sqtdso[4] = (unsigned short )0;

   sqlstm.sqhstv[5] = (         void  \*)&xnatroamom;

   sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[5] = (         int  )0;

   sqlstm.sqindv[5] = (         void  \*)0;

   sqlstm.sqinds[5] = (         int  )0;

   sqlstm.sqharm[5] = (unsigned int  )0;

   sqlstm.sqadto[5] = (unsigned short )0;

   sqlstm.sqtdso[5] = (unsigned short )0;

   sqlstm.sqhstv[6] = (         void  \*)&xisdmin;

   sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[6] = (         int  )0;

   sqlstm.sqindv[6] = (         void  \*)0;

   sqlstm.sqinds[6] = (         int  )0;

   sqlstm.sqharm[6] = (unsigned int  )0;

   sqlstm.sqadto[6] = (unsigned short )0;

   sqlstm.sqtdso[6] = (unsigned short )0;

   sqlstm.sqhstv[7] = (         void  \*)&xinterroami;

   sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[7] = (         int  )0;

   sqlstm.sqindv[7] = (         void  \*)0;

   sqlstm.sqinds[7] = (         int  )0;

   sqlstm.sqharm[7] = (unsigned int  )0;

   sqlstm.sqadto[7] = (unsigned short )0;

   sqlstm.sqtdso[7] = (unsigned short )0;

   sqlstm.sqhstv[8] = (         void  \*)&xinterroamo;

   sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[8] = (         int  )0;

   sqlstm.sqindv[8] = (         void  \*)0;

   sqlstm.sqinds[8] = (         int  )0;

   sqlstm.sqharm[8] = (unsigned int  )0;

   sqlstm.sqadto[8] = (unsigned short )0;

   sqlstm.sqtdso[8] = (unsigned short )0;

   sqlstm.sqhstv[9] = (         void  \*)&xbill;

   sqlstm.sqhstl[9] = (unsigned int  )sizeof(int);

   sqlstm.sqhsts[9] = (         int  )0;

   sqlstm.sqindv[9] = (         void  \*)0;

   sqlstm.sqinds[9] = (         int  )0;

   sqlstm.sqharm[9] = (unsigned int  )0;

   sqlstm.sqadto[9] = (unsigned short )0;

   sqlstm.sqtdso[9] = (unsigned short )0;

   sqlstm.sqphsv = sqlstm.sqhstv;

   sqlstm.sqphsl = sqlstm.sqhstl;

   sqlstm.sqphss = sqlstm.sqhsts;

   sqlstm.sqpind = sqlstm.sqindv;

   sqlstm.sqpins = sqlstm.sqinds;

   sqlstm.sqparm = sqlstm.sqharm;

   sqlstm.sqparc = sqlstm.sqharc;

   sqlstm.sqpadto = sqlstm.sqadto;

   sqlstm.sqptdso = sqlstm.sqtdso;

   sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        /\* exec sql insert into tcal (SMSR,MMSR,GPRSRATE,NATROAMIR,NATROAMOR,STDRATE,INTERROAMIR,INTERROAMOR,ISDRATE,GRANDTOTAL,CALL\_ID,BANK\_ID) values (:xsmsr,:xmmsr,:xgprsrate,:xnatroamir,:xnatroamor,:xstdrate,:xinterroamir,:xinterroamor,:xisdrate,:xgrandtotal,:xcall,:xbank); \*/

{

   struct sqlexd sqlstm;

   sqlstm.sqlvsn = 12;

   sqlstm.arrsiz = 12;

   sqlstm.sqladtp = &sqladt;

   sqlstm.sqltdsp = &sqltds;

   sqlstm.stmt = "insert into tcal (SMSR,MMSR,GPRSRATE,NATROAMIR,NATROAMOR,S\

TDRATE,INTERROAMIR,INTERROAMOR,ISDRATE,GRANDTOTAL,CALL\_ID,BANK\_ID) values (:b0\

,:b1,:b2,:b3,:b4,:b5,:b6,:b7,:b8,:b9,:b10,:b11)";

   sqlstm.iters = (unsigned int  )1;

   sqlstm.offset = (unsigned int  )753;

   sqlstm.cud = sqlcud0;

   sqlstm.sqlest = (unsigned char  \*)&sqlca;

   sqlstm.sqlety = (unsigned short)4352;

   sqlstm.occurs = (unsigned int  )0;

   sqlstm.sqhstv[0] = (         void  \*)&xsmsr;

   sqlstm.sqhstl[0] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[0] = (         int  )0;

   sqlstm.sqindv[0] = (         void  \*)0;

   sqlstm.sqinds[0] = (         int  )0;

   sqlstm.sqharm[0] = (unsigned int  )0;

   sqlstm.sqadto[0] = (unsigned short )0;

   sqlstm.sqtdso[0] = (unsigned short )0;

   sqlstm.sqhstv[1] = (         void  \*)&xmmsr;

   sqlstm.sqhstl[1] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[1] = (         int  )0;

   sqlstm.sqindv[1] = (         void  \*)0;

   sqlstm.sqinds[1] = (         int  )0;

   sqlstm.sqharm[1] = (unsigned int  )0;

   sqlstm.sqadto[1] = (unsigned short )0;

   sqlstm.sqtdso[1] = (unsigned short )0;

   sqlstm.sqhstv[2] = (         void  \*)&xgprsrate;

   sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[2] = (         int  )0;

   sqlstm.sqindv[2] = (         void  \*)0;

   sqlstm.sqinds[2] = (         int  )0;

   sqlstm.sqharm[2] = (unsigned int  )0;

   sqlstm.sqadto[2] = (unsigned short )0;

   sqlstm.sqtdso[2] = (unsigned short )0;

   sqlstm.sqhstv[3] = (         void  \*)&xnatroamir;

   sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[3] = (         int  )0;

   sqlstm.sqindv[3] = (         void  \*)0;

   sqlstm.sqinds[3] = (         int  )0;

   sqlstm.sqharm[3] = (unsigned int  )0;

   sqlstm.sqadto[3] = (unsigned short )0;

   sqlstm.sqtdso[3] = (unsigned short )0;

   sqlstm.sqhstv[4] = (         void  \*)&xnatroamor;

   sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[4] = (         int  )0;

   sqlstm.sqindv[4] = (         void  \*)0;

   sqlstm.sqinds[4] = (         int  )0;

   sqlstm.sqharm[4] = (unsigned int  )0;

   sqlstm.sqadto[4] = (unsigned short )0;

   sqlstm.sqtdso[4] = (unsigned short )0;

   sqlstm.sqhstv[5] = (         void  \*)&xstdrate;

   sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[5] = (         int  )0;

   sqlstm.sqindv[5] = (         void  \*)0;

   sqlstm.sqinds[5] = (         int  )0;

   sqlstm.sqharm[5] = (unsigned int  )0;

   sqlstm.sqadto[5] = (unsigned short )0;

   sqlstm.sqtdso[5] = (unsigned short )0;

   sqlstm.sqhstv[6] = (         void  \*)&xinterroamir;

   sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[6] = (         int  )0;

   sqlstm.sqindv[6] = (         void  \*)0;

   sqlstm.sqinds[6] = (         int  )0;

   sqlstm.sqharm[6] = (unsigned int  )0;

   sqlstm.sqadto[6] = (unsigned short )0;

   sqlstm.sqtdso[6] = (unsigned short )0;

   sqlstm.sqhstv[7] = (         void  \*)&xinterroamor;

   sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[7] = (         int  )0;

   sqlstm.sqindv[7] = (         void  \*)0;

   sqlstm.sqinds[7] = (         int  )0;

   sqlstm.sqharm[7] = (unsigned int  )0;

   sqlstm.sqadto[7] = (unsigned short )0;

   sqlstm.sqtdso[7] = (unsigned short )0;

   sqlstm.sqhstv[8] = (         void  \*)&xisdrate;

   sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[8] = (         int  )0;

   sqlstm.sqindv[8] = (         void  \*)0;

   sqlstm.sqinds[8] = (         int  )0;

   sqlstm.sqharm[8] = (unsigned int  )0;

   sqlstm.sqadto[8] = (unsigned short )0;

   sqlstm.sqtdso[8] = (unsigned short )0;

   sqlstm.sqhstv[9] = (         void  \*)&xgrandtotal;

   sqlstm.sqhstl[9] = (unsigned int  )sizeof(float);

   sqlstm.sqhsts[9] = (         int  )0;

   sqlstm.sqindv[9] = (         void  \*)0;

   sqlstm.sqinds[9] = (         int  )0;

   sqlstm.sqharm[9] = (unsigned int  )0;

   sqlstm.sqadto[9] = (unsigned short )0;

   sqlstm.sqtdso[9] = (unsigned short )0;

   sqlstm.sqhstv[10] = (         void  \*)&xcall;

   sqlstm.sqhstl[10] = (unsigned int  )sizeof(int);

   sqlstm.sqhsts[10] = (         int  )0;

   sqlstm.sqindv[10] = (         void  \*)0;

   sqlstm.sqinds[10] = (         int  )0;

   sqlstm.sqharm[10] = (unsigned int  )0;

   sqlstm.sqadto[10] = (unsigned short )0;

   sqlstm.sqtdso[10] = (unsigned short )0;

   sqlstm.sqhstv[11] = (         void  \*)&xbank;

   sqlstm.sqhstl[11] = (unsigned int  )sizeof(int);

   sqlstm.sqhsts[11] = (         int  )0;

   sqlstm.sqindv[11] = (         void  \*)0;

   sqlstm.sqinds[11] = (         int  )0;

   sqlstm.sqharm[11] = (unsigned int  )0;

   sqlstm.sqadto[11] = (unsigned short )0;

   sqlstm.sqtdso[11] = (unsigned short )0;

   sqlstm.sqphsv = sqlstm.sqhstv;

   sqlstm.sqphsl = sqlstm.sqhstl;

   sqlstm.sqphss = sqlstm.sqhsts;

   sqlstm.sqpind = sqlstm.sqindv;

   sqlstm.sqpins = sqlstm.sqinds;

   sqlstm.sqparm = sqlstm.sqharm;

   sqlstm.sqparc = sqlstm.sqharc;

   sqlstm.sqpadto = sqlstm.sqadto;

   sqlstm.sqptdso = sqlstm.sqtdso;

   sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

        if(sqlca.sqlcode==0)

            cout<<"success";

        else

            cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

        /\* exec sql commit work ; \*/

{

        struct sqlexd sqlstm;

        sqlstm.sqlvsn = 12;

        sqlstm.arrsiz = 12;

        sqlstm.sqladtp = &sqladt;

        sqlstm.sqltdsp = &sqltds;

        sqlstm.iters = (unsigned int  )1;

        sqlstm.offset = (unsigned int  )816;

        sqlstm.cud = sqlcud0;

        sqlstm.sqlest = (unsigned char  \*)&sqlca;

        sqlstm.sqlety = (unsigned short)4352;

        sqlstm.occurs = (unsigned int  )0;

        sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

     }

     else

     {

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

        cout<<"\n\nEnter any key:";

        cin.get();

     }

}

void selectbill(char p[10],int a)

{

    for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xphone[10]=NULL;

    xacc=a;

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (PH\

ONE\_NO=:b2 and ACC\_NO=:b3)";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )831;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xbill;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xcall;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)xphone;

  sqlstm.sqhstl[2] = (unsigned int  )12;

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xacc;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql select SMS,MMS,GPRSKP,STDMIN,NATROAMIM,NATROAMOM,ISDMIN,INTERROAMI,INTERROAMO into :xsms,:xmms,:xgprskb,:xstdmin,:xnatroamim,:xnatroamom,:xisdmin,:xinterroami,:xinterroamo from Tbill where bill\_id=:xbill ; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select SMS ,MMS ,GPRSKP ,STDMIN ,NATROAMIM ,NATROAMOM ,ISDM\

IN ,INTERROAMI ,INTERROAMO into :b0,:b1,:b2,:b3,:b4,:b5,:b6,:b7,:b8  from Tbil\

l where bill\_id=:b9";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )862;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xsms;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xmms;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)&xgprskb;

  sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xstdmin;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqhstv[4] = (         void  \*)&xnatroamim;

  sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[4] = (         int  )0;

  sqlstm.sqindv[4] = (         void  \*)0;

  sqlstm.sqinds[4] = (         int  )0;

  sqlstm.sqharm[4] = (unsigned int  )0;

  sqlstm.sqadto[4] = (unsigned short )0;

  sqlstm.sqtdso[4] = (unsigned short )0;

  sqlstm.sqhstv[5] = (         void  \*)&xnatroamom;

  sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[5] = (         int  )0;

  sqlstm.sqindv[5] = (         void  \*)0;

  sqlstm.sqinds[5] = (         int  )0;

  sqlstm.sqharm[5] = (unsigned int  )0;

  sqlstm.sqadto[5] = (unsigned short )0;

  sqlstm.sqtdso[5] = (unsigned short )0;

  sqlstm.sqhstv[6] = (         void  \*)&xisdmin;

  sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[6] = (         int  )0;

  sqlstm.sqindv[6] = (         void  \*)0;

  sqlstm.sqinds[6] = (         int  )0;

  sqlstm.sqharm[6] = (unsigned int  )0;

  sqlstm.sqadto[6] = (unsigned short )0;

  sqlstm.sqtdso[6] = (unsigned short )0;

  sqlstm.sqhstv[7] = (         void  \*)&xinterroami;

  sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[7] = (         int  )0;

  sqlstm.sqindv[7] = (         void  \*)0;

  sqlstm.sqinds[7] = (         int  )0;

  sqlstm.sqharm[7] = (unsigned int  )0;

  sqlstm.sqadto[7] = (unsigned short )0;

  sqlstm.sqtdso[7] = (unsigned short )0;

  sqlstm.sqhstv[8] = (         void  \*)&xinterroamo;

  sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[8] = (         int  )0;

  sqlstm.sqindv[8] = (         void  \*)0;

  sqlstm.sqinds[8] = (         int  )0;

  sqlstm.sqharm[8] = (unsigned int  )0;

  sqlstm.sqadto[8] = (unsigned short )0;

  sqlstm.sqtdso[8] = (unsigned short )0;

  sqlstm.sqhstv[9] = (         void  \*)&xbill;

  sqlstm.sqhstl[9] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[9] = (         int  )0;

  sqlstm.sqindv[9] = (         void  \*)0;

  sqlstm.sqinds[9] = (         int  )0;

  sqlstm.sqharm[9] = (unsigned int  )0;

  sqlstm.sqadto[9] = (unsigned short )0;

  sqlstm.sqtdso[9] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql select SMSR,MMSR,GPRSRATE,NATROAMIR,NATROAMOR,STDRATE,INTERROAMIR,INTERROAMOR,ISDRATE,GRANDTOTAL into :xsmsr,:xmmsr,:xgprsrate,:xnatroamir,:xnatroamor,:xstdrate,:xinterroamir,:xinterroamor,:xisdrate,:xgrandtotal from tcal where  CALL\_ID=:xcall; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "select SMSR ,MMSR ,GPRSRATE ,NATROAMIR ,NATROAMOR ,STDRATE \

,INTERROAMIR ,INTERROAMOR ,ISDRATE ,GRANDTOTAL into :b0,:b1,:b2,:b3,:b4,:b5,:b\

6,:b7,:b8,:b9  from tcal where CALL\_ID=:b10";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )917;

  sqlstm.selerr = (unsigned short)1;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xsmsr;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xmmsr;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)&xgprsrate;

  sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xnatroamir;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqhstv[4] = (         void  \*)&xnatroamor;

  sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[4] = (         int  )0;

  sqlstm.sqindv[4] = (         void  \*)0;

  sqlstm.sqinds[4] = (         int  )0;

  sqlstm.sqharm[4] = (unsigned int  )0;

  sqlstm.sqadto[4] = (unsigned short )0;

  sqlstm.sqtdso[4] = (unsigned short )0;

  sqlstm.sqhstv[5] = (         void  \*)&xstdrate;

  sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[5] = (         int  )0;

  sqlstm.sqindv[5] = (         void  \*)0;

  sqlstm.sqinds[5] = (         int  )0;

  sqlstm.sqharm[5] = (unsigned int  )0;

  sqlstm.sqadto[5] = (unsigned short )0;

  sqlstm.sqtdso[5] = (unsigned short )0;

  sqlstm.sqhstv[6] = (         void  \*)&xinterroamir;

  sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[6] = (         int  )0;

  sqlstm.sqindv[6] = (         void  \*)0;

  sqlstm.sqinds[6] = (         int  )0;

  sqlstm.sqharm[6] = (unsigned int  )0;

  sqlstm.sqadto[6] = (unsigned short )0;

  sqlstm.sqtdso[6] = (unsigned short )0;

  sqlstm.sqhstv[7] = (         void  \*)&xinterroamor;

  sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[7] = (         int  )0;

  sqlstm.sqindv[7] = (         void  \*)0;

  sqlstm.sqinds[7] = (         int  )0;

  sqlstm.sqharm[7] = (unsigned int  )0;

  sqlstm.sqadto[7] = (unsigned short )0;

  sqlstm.sqtdso[7] = (unsigned short )0;

  sqlstm.sqhstv[8] = (         void  \*)&xisdrate;

  sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[8] = (         int  )0;

  sqlstm.sqindv[8] = (         void  \*)0;

  sqlstm.sqinds[8] = (         int  )0;

  sqlstm.sqharm[8] = (unsigned int  )0;

  sqlstm.sqadto[8] = (unsigned short )0;

  sqlstm.sqtdso[8] = (unsigned short )0;

  sqlstm.sqhstv[9] = (         void  \*)&xgrandtotal;

  sqlstm.sqhstl[9] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[9] = (         int  )0;

  sqlstm.sqindv[9] = (         void  \*)0;

  sqlstm.sqinds[9] = (         int  )0;

  sqlstm.sqharm[9] = (unsigned int  )0;

  sqlstm.sqadto[9] = (unsigned short )0;

  sqlstm.sqtdso[9] = (unsigned short )0;

  sqlstm.sqhstv[10] = (         void  \*)&xcall;

  sqlstm.sqhstl[10] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[10] = (         int  )0;

  sqlstm.sqindv[10] = (         void  \*)0;

  sqlstm.sqinds[10] = (         int  )0;

  sqlstm.sqharm[10] = (unsigned int  )0;

  sqlstm.sqadto[10] = (unsigned short )0;

  sqlstm.sqtdso[10] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

        cout<<"=============================================================================";

        cout<<endl;

        cout<<"\t\t        CALLS          ";

        cout<<"\t"<<"      MINUTES/No.";

        cout<<"\t"<<"AMOUNT";

        cout<<endl;

        cout<<"=============================================================================";

        cout<<endl<<"\t\t  Incoming Calls          ";

        cout<<"\t"<<xstdmin;

        cout<<"\t\t"<<xstdrate;

        cout<<endl<<"\t\t  No of SMS               ";

        cout<<"\t"<<xsms;

        cout<<"\t\t"<<xsmsr;

        cout<<endl<<"\t\t  Roaming Incoming Calls   ";

        cout<<"\t"<<xnatroamim;

        cout<<"\t\t"<<xnatroamir;

        cout<<endl<<"\t\t  Roaming Outgoing Calls   ";

        cout<<"\t"<<xnatroamom;

        cout<<"\t\t"<<xnatroamor;

        cout<<endl<<endl<<"\t\t  GPRS Usage ( in Kb )     ";

        cout<<"\t"<<xgprskb;

        cout<<"\t\t"<<xgprsrate;

        cout<<endl<<"\t\t  No of MMS                ";

        cout<<"\t"<<xmms;

        cout<<"\t\t"<<xmmsr;

        cout<<endl<<endl<<"\t\t  ISD Calls                ";

        cout<<"\t"<<xisdmin;

        cout<<"\t\t"<<xisdrate;

        cout<<endl<<"\t\t  Inter Roaming Incoming   ";

        cout<<"\t"<<xinterroami;

        cout<<"\t\t"<<xinterroamir;

        cout<<endl<<"\t\t  Inter Roaming Outgoing   ";

        cout<<"\t"<<xinterroamo;

        cout<<"\t\t"<<xinterroamor;

        cout<<endl<<endl<<"\t\t  | Total Charges |           ";

        cout<<"\t\t\t"<<xgrandtotal;

    }

    else

    {

            cout<<endl<<endl<<"============================================================================="<<endl;

            cout<<"\t\t\tFile not found ....No such user"<<endl;

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl;

    }

}

void editbill(char p[10],int a)

{

    char d;

    system("cls");

    for(int i=0;i<10;i++)

        xphone[i]=p[i];

    xacc=a;

    cout<<"=============================================================================";

    cout<<"\n\t\t\t Current Customer Details";

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

    selectbill(xphone,xacc);

    cout<<endl<<"========================================================================";

    cout<<endl<<"\t                 Press Any Key To Continue      "<<endl;

    cout<<endl<<"========================================================================";

    cin>>d;

    system("cls");

    cout<<endl<<"========================================================================";

    cout<<endl<<"                     Edit What?????      ";

    cout<<endl<<"========================================================================";

    cout<<endl;

    cout<<"1. Incoming Calls\t\t\t6. No of MMS"<<endl;

    cout<<"2. No of SMS\t\t\t\t7. ISD Calls\t\t"<<endl;

    cout<<"3. Roaming Incoming Calls\t\t8. International Roaming Incoming"<<endl;

    cout<<"4. Roaming Outgoing Calls\t\t9. International Roaming Outgoing"<<endl;

    cout<<"5. GPRS Usage ( in Kb )\t\t\t"<<endl;

    cout<<"=============================================================================";

    cout<<endl<<"Enter  Choice      : ";

    int ch;

    pp :

    cin>>ch;

    if(ch==1)

    {

        cout<<"Enter Incoming Calls min    : ";

        cin>>xstdmin;

    }

    if(ch==2)

    {

        cout<<"Enter SMS    : ";

        cin>>xsms;

    }

    if(ch==3)

    {

        cout<<"Enter Roaming in     : ";

        cin>>xnatroamim;

    }

    if(ch==4)

    {

        cout<<"Enter Roaming out     : ";

        cin>>xnatroamom;

    }

    if(ch==5)

    {

        cout<<"Enter GPRS Usage    : ";

        cin>>xgprskb;

    }

    if(ch==6)

    {

        cout<<"Enter MMS no     : ";

        cin>>xmms;

    }

    if(ch==7)

    {

        cout<<"Enter ISD Min    : ";

        cin>>xisdmin;

    }

    if(ch==8)

    {

        cout<<"Enter ISD Roam in    : ";

        cin>>xinterroami;

    }

    if(ch==9)

    {

        cout<<"Enter ISD Roamout    : ";

        cin>>xinterroamo;

    }

    if((ch>9)||(ch==0))

    {

        cout<<"\n\t\t\t\tInvalid Choice !!!!!"<<endl<<"\t\t\t\t";

        goto pp;

    }

    cal();

    /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where PHONE\_NO=:xphone AND ACC\_NO=:xacc; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 12;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone where (\

PHONE\_NO=:b2 and ACC\_NO=:b3)";

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )976;

    sqlstm.selerr = (unsigned short)1;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlstm.sqhstv[0] = (         void  \*)&xbill;

    sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[0] = (         int  )0;

    sqlstm.sqindv[0] = (         void  \*)0;

    sqlstm.sqinds[0] = (         int  )0;

    sqlstm.sqharm[0] = (unsigned int  )0;

    sqlstm.sqadto[0] = (unsigned short )0;

    sqlstm.sqtdso[0] = (unsigned short )0;

    sqlstm.sqhstv[1] = (         void  \*)&xcall;

    sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[1] = (         int  )0;

    sqlstm.sqindv[1] = (         void  \*)0;

    sqlstm.sqinds[1] = (         int  )0;

    sqlstm.sqharm[1] = (unsigned int  )0;

    sqlstm.sqadto[1] = (unsigned short )0;

    sqlstm.sqtdso[1] = (unsigned short )0;

    sqlstm.sqhstv[2] = (         void  \*)xphone;

    sqlstm.sqhstl[2] = (unsigned int  )12;

    sqlstm.sqhsts[2] = (         int  )0;

    sqlstm.sqindv[2] = (         void  \*)0;

    sqlstm.sqinds[2] = (         int  )0;

    sqlstm.sqharm[2] = (unsigned int  )0;

    sqlstm.sqadto[2] = (unsigned short )0;

    sqlstm.sqtdso[2] = (unsigned short )0;

    sqlstm.sqhstv[3] = (         void  \*)&xacc;

    sqlstm.sqhstl[3] = (unsigned int  )sizeof(int);

    sqlstm.sqhsts[3] = (         int  )0;

    sqlstm.sqindv[3] = (         void  \*)0;

    sqlstm.sqinds[3] = (         int  )0;

    sqlstm.sqharm[3] = (unsigned int  )0;

    sqlstm.sqadto[3] = (unsigned short )0;

    sqlstm.sqtdso[3] = (unsigned short )0;

    sqlstm.sqphsv = sqlstm.sqhstv;

    sqlstm.sqphsl = sqlstm.sqhstl;

    sqlstm.sqphss = sqlstm.sqhsts;

    sqlstm.sqpind = sqlstm.sqindv;

    sqlstm.sqpins = sqlstm.sqinds;

    sqlstm.sqparm = sqlstm.sqharm;

    sqlstm.sqparc = sqlstm.sqharc;

    sqlstm.sqpadto = sqlstm.sqadto;

    sqlstm.sqptdso = sqlstm.sqtdso;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql update tbill set SMS = :xsms, MMS =:xmms, GPRSKP = :xgprskb, STDMIN = :xstdmin, NATROAMIM = :xnatroamim, NATROAMOM = :xnatroamom, ISDMIN = :xisdmin, INTERROAMI = :xinterroami, INTERROAMO = :xinterroamo where bill\_id=:xbill ; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "update tbill  set SMS=:b0,MMS=:b1,GPRSKP=:b2,STDMIN=:b3,NAT\

ROAMIM=:b4,NATROAMOM=:b5,ISDMIN=:b6,INTERROAMI=:b7,INTERROAMO=:b8 where bill\_i\

d=:b9";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )1007;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xsms;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xmms;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)&xgprskb;

  sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xstdmin;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqhstv[4] = (         void  \*)&xnatroamim;

  sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[4] = (         int  )0;

  sqlstm.sqindv[4] = (         void  \*)0;

  sqlstm.sqinds[4] = (         int  )0;

  sqlstm.sqharm[4] = (unsigned int  )0;

  sqlstm.sqadto[4] = (unsigned short )0;

  sqlstm.sqtdso[4] = (unsigned short )0;

  sqlstm.sqhstv[5] = (         void  \*)&xnatroamom;

  sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[5] = (         int  )0;

  sqlstm.sqindv[5] = (         void  \*)0;

  sqlstm.sqinds[5] = (         int  )0;

  sqlstm.sqharm[5] = (unsigned int  )0;

  sqlstm.sqadto[5] = (unsigned short )0;

  sqlstm.sqtdso[5] = (unsigned short )0;

  sqlstm.sqhstv[6] = (         void  \*)&xisdmin;

  sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[6] = (         int  )0;

  sqlstm.sqindv[6] = (         void  \*)0;

  sqlstm.sqinds[6] = (         int  )0;

  sqlstm.sqharm[6] = (unsigned int  )0;

  sqlstm.sqadto[6] = (unsigned short )0;

  sqlstm.sqtdso[6] = (unsigned short )0;

  sqlstm.sqhstv[7] = (         void  \*)&xinterroami;

  sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[7] = (         int  )0;

  sqlstm.sqindv[7] = (         void  \*)0;

  sqlstm.sqinds[7] = (         int  )0;

  sqlstm.sqharm[7] = (unsigned int  )0;

  sqlstm.sqadto[7] = (unsigned short )0;

  sqlstm.sqtdso[7] = (unsigned short )0;

  sqlstm.sqhstv[8] = (         void  \*)&xinterroamo;

  sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[8] = (         int  )0;

  sqlstm.sqindv[8] = (         void  \*)0;

  sqlstm.sqinds[8] = (         int  )0;

  sqlstm.sqharm[8] = (unsigned int  )0;

  sqlstm.sqadto[8] = (unsigned short )0;

  sqlstm.sqtdso[8] = (unsigned short )0;

  sqlstm.sqhstv[9] = (         void  \*)&xbill;

  sqlstm.sqhstl[9] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[9] = (         int  )0;

  sqlstm.sqindv[9] = (         void  \*)0;

  sqlstm.sqinds[9] = (         int  )0;

  sqlstm.sqharm[9] = (unsigned int  )0;

  sqlstm.sqadto[9] = (unsigned short )0;

  sqlstm.sqtdso[9] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    /\* exec sql update tcal set SMSR = :xsmsr, MMSR = :xmmsr, GPRSRATE = :xgprsrate, NATROAMIR = :xnatroamir, NATROAMOR = :xnatroamor, STDRATE = :xstdrate, INTERROAMIR =:xinterroamir, INTERROAMOR =:xinterroamor, ISDRATE =:xisdrate, GRANDTOTAL = :xgrandtotal where  CALL\_ID=:xcall; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.stmt = "update tcal  set SMSR=:b0,MMSR=:b1,GPRSRATE=:b2,NATROAMIR=:\

b3,NATROAMOR=:b4,STDRATE=:b5,INTERROAMIR=:b6,INTERROAMOR=:b7,ISDRATE=:b8,GRAND\

TOTAL=:b9 where CALL\_ID=:b10";

  sqlstm.iters = (unsigned int  )1;

  sqlstm.offset = (unsigned int  )1062;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)&xsmsr;

  sqlstm.sqhstl[0] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[0] = (         int  )0;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)&xmmsr;

  sqlstm.sqhstl[1] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[1] = (         int  )0;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqhstv[2] = (         void  \*)&xgprsrate;

  sqlstm.sqhstl[2] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[2] = (         int  )0;

  sqlstm.sqindv[2] = (         void  \*)0;

  sqlstm.sqinds[2] = (         int  )0;

  sqlstm.sqharm[2] = (unsigned int  )0;

  sqlstm.sqadto[2] = (unsigned short )0;

  sqlstm.sqtdso[2] = (unsigned short )0;

  sqlstm.sqhstv[3] = (         void  \*)&xnatroamir;

  sqlstm.sqhstl[3] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[3] = (         int  )0;

  sqlstm.sqindv[3] = (         void  \*)0;

  sqlstm.sqinds[3] = (         int  )0;

  sqlstm.sqharm[3] = (unsigned int  )0;

  sqlstm.sqadto[3] = (unsigned short )0;

  sqlstm.sqtdso[3] = (unsigned short )0;

  sqlstm.sqhstv[4] = (         void  \*)&xnatroamor;

  sqlstm.sqhstl[4] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[4] = (         int  )0;

  sqlstm.sqindv[4] = (         void  \*)0;

  sqlstm.sqinds[4] = (         int  )0;

  sqlstm.sqharm[4] = (unsigned int  )0;

  sqlstm.sqadto[4] = (unsigned short )0;

  sqlstm.sqtdso[4] = (unsigned short )0;

  sqlstm.sqhstv[5] = (         void  \*)&xstdrate;

  sqlstm.sqhstl[5] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[5] = (         int  )0;

  sqlstm.sqindv[5] = (         void  \*)0;

  sqlstm.sqinds[5] = (         int  )0;

  sqlstm.sqharm[5] = (unsigned int  )0;

  sqlstm.sqadto[5] = (unsigned short )0;

  sqlstm.sqtdso[5] = (unsigned short )0;

  sqlstm.sqhstv[6] = (         void  \*)&xinterroamir;

  sqlstm.sqhstl[6] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[6] = (         int  )0;

  sqlstm.sqindv[6] = (         void  \*)0;

  sqlstm.sqinds[6] = (         int  )0;

  sqlstm.sqharm[6] = (unsigned int  )0;

  sqlstm.sqadto[6] = (unsigned short )0;

  sqlstm.sqtdso[6] = (unsigned short )0;

  sqlstm.sqhstv[7] = (         void  \*)&xinterroamor;

  sqlstm.sqhstl[7] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[7] = (         int  )0;

  sqlstm.sqindv[7] = (         void  \*)0;

  sqlstm.sqinds[7] = (         int  )0;

  sqlstm.sqharm[7] = (unsigned int  )0;

  sqlstm.sqadto[7] = (unsigned short )0;

  sqlstm.sqtdso[7] = (unsigned short )0;

  sqlstm.sqhstv[8] = (         void  \*)&xisdrate;

  sqlstm.sqhstl[8] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[8] = (         int  )0;

  sqlstm.sqindv[8] = (         void  \*)0;

  sqlstm.sqinds[8] = (         int  )0;

  sqlstm.sqharm[8] = (unsigned int  )0;

  sqlstm.sqadto[8] = (unsigned short )0;

  sqlstm.sqtdso[8] = (unsigned short )0;

  sqlstm.sqhstv[9] = (         void  \*)&xgrandtotal;

  sqlstm.sqhstl[9] = (unsigned int  )sizeof(float);

  sqlstm.sqhsts[9] = (         int  )0;

  sqlstm.sqindv[9] = (         void  \*)0;

  sqlstm.sqinds[9] = (         int  )0;

  sqlstm.sqharm[9] = (unsigned int  )0;

  sqlstm.sqadto[9] = (unsigned short )0;

  sqlstm.sqtdso[9] = (unsigned short )0;

  sqlstm.sqhstv[10] = (         void  \*)&xcall;

  sqlstm.sqhstl[10] = (unsigned int  )sizeof(int);

  sqlstm.sqhsts[10] = (         int  )0;

  sqlstm.sqindv[10] = (         void  \*)0;

  sqlstm.sqinds[10] = (         int  )0;

  sqlstm.sqharm[10] = (unsigned int  )0;

  sqlstm.sqadto[10] = (unsigned short )0;

  sqlstm.sqtdso[10] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            if(sqlca.sqlcode==0)

    {

        cout<<"Success"<<endl;

    }

    else

        cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc<<endl<<"File not found";

    /\* exec sql commit work; \*/

{

    struct sqlexd sqlstm;

    sqlstm.sqlvsn = 12;

    sqlstm.arrsiz = 12;

    sqlstm.sqladtp = &sqladt;

    sqlstm.sqltdsp = &sqltds;

    sqlstm.iters = (unsigned int  )1;

    sqlstm.offset = (unsigned int  )1121;

    sqlstm.cud = sqlcud0;

    sqlstm.sqlest = (unsigned char  \*)&sqlca;

    sqlstm.sqlety = (unsigned short)4352;

    sqlstm.occurs = (unsigned int  )0;

    sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

}

int main()

{

    int ch;

    char r,tem;

    char uname[30],pw[10];

    gotoxy(1,5);

    cout<<"\t\t\t";

    cout<<"=============================================================================";

    gotoxy(32,6);

    cout<<"\t\t\t";

    cout<<"HOMEPAGE OF ADMIN";

    cout<<endl;

    cout<<"\t\t\t";

    cout<<"=============================================================================";

    gotoxy(27,10);

    cout<<"Enter username: ";

    cin>>uname;

    if(strcmp(uname,"admin")==0)

    {

        g:

        cout<<"\t\t\t   Enter password: ";

        cin>>pw;

        if(strcmp(pw,"admin199901")==0)

        {

        system("cls");

        }

        else

        {

        cout<<"Wrong password";

        goto g;

        }

    }

    else

    {

        cout<<"wrong username";

        exit(0);

    }

    y:

    system("cls");

    /\* exec sql connect:u identified by:p; \*/

{

  struct sqlexd sqlstm;

  sqlstm.sqlvsn = 12;

  sqlstm.arrsiz = 12;

  sqlstm.sqladtp = &sqladt;

  sqlstm.sqltdsp = &sqltds;

  sqlstm.iters = (unsigned int  )10;

  sqlstm.offset = (unsigned int  )1136;

  sqlstm.cud = sqlcud0;

  sqlstm.sqlest = (unsigned char  \*)&sqlca;

  sqlstm.sqlety = (unsigned short)4352;

  sqlstm.occurs = (unsigned int  )0;

  sqlstm.sqhstv[0] = (         void  \*)u;

  sqlstm.sqhstl[0] = (unsigned int  )20;

  sqlstm.sqhsts[0] = (         int  )20;

  sqlstm.sqindv[0] = (         void  \*)0;

  sqlstm.sqinds[0] = (         int  )0;

  sqlstm.sqharm[0] = (unsigned int  )0;

  sqlstm.sqadto[0] = (unsigned short )0;

  sqlstm.sqtdso[0] = (unsigned short )0;

  sqlstm.sqhstv[1] = (         void  \*)p;

  sqlstm.sqhstl[1] = (unsigned int  )20;

  sqlstm.sqhsts[1] = (         int  )20;

  sqlstm.sqindv[1] = (         void  \*)0;

  sqlstm.sqinds[1] = (         int  )0;

  sqlstm.sqharm[1] = (unsigned int  )0;

  sqlstm.sqadto[1] = (unsigned short )0;

  sqlstm.sqtdso[1] = (unsigned short )0;

  sqlstm.sqphsv = sqlstm.sqhstv;

  sqlstm.sqphsl = sqlstm.sqhstl;

  sqlstm.sqphss = sqlstm.sqhsts;

  sqlstm.sqpind = sqlstm.sqindv;

  sqlstm.sqpins = sqlstm.sqinds;

  sqlstm.sqparm = sqlstm.sqharm;

  sqlstm.sqparc = sqlstm.sqharc;

  sqlstm.sqpadto = sqlstm.sqadto;

  sqlstm.sqptdso = sqlstm.sqtdso;

  sqlstm.sqlcmax = (unsigned int )100;

  sqlstm.sqlcmin = (unsigned int )2;

  sqlstm.sqlcincr = (unsigned int )1;

  sqlstm.sqlctimeout = (unsigned int )0;

  sqlstm.sqlcnowait = (unsigned int )0;

  sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

    if(sqlca.sqlcode==0)

    {

    cout<<"\n\n\t\t";

    cout<<"============================================================================="<<endl<<endl;

    cout<<"\t\t";

    cout<<"\t\t\tTELEPHONE BILLER"<<endl;

    cout<<endl;

    cout<<"\t\t";

    cout<<"============================================================================="<<endl<<endl;

    cout<<"\t\t";

    cout<<"1) ENTER CUSTOMER DATABASE"<<endl<<endl;

    cout<<"\t\t";

    cout<<"2) ENTER PHONE DATABASE"<<endl<<endl;

    cout<<"\t\t";

    cout<<"3) ENTER BILL  DATABASE"<<endl<<endl;

    cout<<"\t\t";

    cout<<"4) ENTER  BANK DATABASE"<<endl<<endl;

    cout<<"\t\t";

    cout<<"5) ENTER  BILLING  DETAILS"<<endl<<endl;

    cout<<"\t\t";

    cout<<"6) EXIT  FROM  PROGRAM"<<endl<<endl<<endl;

    cout<<"\t\t";

    cout<<"=============================================================================";

    cout<<endl;

    cout<<"\t\t";

    cout<<"        Enter Your Choice     : ";

    cin>>ch;

    switch(ch)

    {

        case 1:

      {

        system("cls");

        int op=0;

        x:

        system("cls");

        cout<<endl<<endl;

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

        cout<<endl;

        cout<<"\t\t\t    CUSTOMER SECTION";

        cout<<endl;

        cout<<"\t\t\t    ================";

        cout<<endl<<endl<<endl;

        cout<<"\t\t\t\t1) ADD\n\n";//over

        cout<<"\t\t\t\t2) EDIT\n\n";

        cout<<"\t\t\t\t3) DELETE\n\n";

        cout<<"\t\t\t\t4) VIEW  \n\n";

        cout<<"\t\t\t\t5) BACK to MAIN";

        cout<<endl<<endl<<endl;

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

        cout<<endl<<"        Enter Your Choice     : ";

        cin>>op;

        if(op==1)

        {

            system("cls");

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tYOU HAVE CHOSEN TO ADD CUSTOMER";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<endl;

            insertcust();

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tCUSTOMER ADDED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key....";

            cin>>r;

            goto x;

        }

        if(op==2)

        {

            system("cls");

            cout<<"============================================================================="<<endl;

            cout<<"\t\t\tYou Have chosen to EDIT CUSTOMER";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\n\n\n\n\n\n\n\n\n\t\t    Enter CUSTOMER id to be EDITED     : ";

            cin>>cid;

            editcust(cid);

            cout<<"=============================================================================";

            cout<<"\n\t\t\tCUSTOMER EDITED SUCCESSFULLY";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl<<"Enter any key....";

            cin>>r;

            goto x;

        }

        if(op==3)

        {

            system("cls");

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tYOU HAVE CHOSEN TO DELETE CUSTOMER";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\t\t\tEnter the account number to delete :";

            cin>>acc;

            /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where ACC\_NO=:acc; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone\

 where ACC\_NO=:b2";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1167;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqhstv[2] = (         void  \*)&acc;

            sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[2] = (         int  )0;

            sqlstm.sqindv[2] = (         void  \*)0;

            sqlstm.sqinds[2] = (         int  )0;

            sqlstm.sqharm[2] = (unsigned int  )0;

            sqlstm.sqadto[2] = (unsigned short )0;

            sqlstm.sqtdso[2] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=\

:b1";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1194;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from bank where BANK\_ID=:xbank; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from bank  where BANK\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1217;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tbill where BILL\_ID=:xbill; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tbill  where BILL\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1236;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tcal  where CALL\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1255;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xcall;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tphone where ACC\_NO=:acc; \*/

{

         struct sqlexd sqlstm;

         sqlstm.sqlvsn = 12;

         sqlstm.arrsiz = 12;

         sqlstm.sqladtp = &sqladt;

         sqlstm.sqltdsp = &sqltds;

         sqlstm.stmt = "delete  from tphone  where ACC\_NO=:b0";

         sqlstm.iters = (unsigned int  )1;

         sqlstm.offset = (unsigned int  )1274;

         sqlstm.cud = sqlcud0;

         sqlstm.sqlest = (unsigned char  \*)&sqlca;

         sqlstm.sqlety = (unsigned short)4352;

         sqlstm.occurs = (unsigned int  )0;

         sqlstm.sqhstv[0] = (         void  \*)&acc;

         sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

         sqlstm.sqhsts[0] = (         int  )0;

         sqlstm.sqindv[0] = (         void  \*)0;

         sqlstm.sqinds[0] = (         int  )0;

         sqlstm.sqharm[0] = (unsigned int  )0;

         sqlstm.sqadto[0] = (unsigned short )0;

         sqlstm.sqtdso[0] = (unsigned short )0;

         sqlstm.sqphsv = sqlstm.sqhstv;

         sqlstm.sqphsl = sqlstm.sqhstl;

         sqlstm.sqphss = sqlstm.sqhsts;

         sqlstm.sqpind = sqlstm.sqindv;

         sqlstm.sqpins = sqlstm.sqinds;

         sqlstm.sqparm = sqlstm.sqharm;

         sqlstm.sqparc = sqlstm.sqharc;

         sqlstm.sqpadto = sqlstm.sqadto;

         sqlstm.sqptdso = sqlstm.sqtdso;

         sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tcustomer where ACC\_NO=:acc; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tcustomer  where ACC\_NO=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1293;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&acc;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql commit work; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1312;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tCUSTOMER DELETED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key....";

            cin>>r;

            goto x;

        }

        if(op==4)

        {

            system("cls");

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have chosen to VIEW CUSTOMER";

            cout<<endl<<endl;

            cout<<"=============================================================================";

            selectcust();

            cout<<endl<<"Enter any key...";

            cin>>r;

            system("cls");

            goto x;

        }

        if(op==5)

        {

            system("cls");

            goto y;

        }

           if(op==0||op>5)

        {

            cout<<"Invalid choice..... Enter any Key";

            cin>>tem;

            goto x;

        }

      }

        case 2:

      {

        system("cls");

        int ck;

        q:

        system("cls");

        cout<<endl<<endl;

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

        cout<<endl;

        cout<<"\t\t\t\tPHONE  SECTION";

        cout<<endl;

        cout<<"\t\t\t\t================";

        cout<<endl<<endl<<endl;

        cout<<"\t\t\t\t1) ADD   \n\n";

        cout<<"\t\t\t\t2) DELETE\n\n";

        cout<<"\t\t\t\t3) SHOW  \n\n";

        cout<<"\t\t\t\t4) BACK to MAIN";

        cout<<endl<<endl<<endl;

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

        cout<<endl<<"        ENTER YOUR CHOICE      : ";

        cin>>ck;

        if(ck==1)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To ADD a PHONE NUMBER ";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            insertphone(cid);

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tPHONE NUMBER ADDED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto q;

        }

        if(ck==2)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To DELETE a PHONE NUMBER ";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\t\tEnter CUSTOMER PHONE NUMBER you want to delete    : ";

            cin>>cphone;

            /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where ACC\_NO=:cid; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone\

 where ACC\_NO=:b2";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1327;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqhstv[2] = (         void  \*)&cid;

            sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[2] = (         int  )0;

            sqlstm.sqindv[2] = (         void  \*)0;

            sqlstm.sqinds[2] = (         int  )0;

            sqlstm.sqharm[2] = (unsigned int  )0;

            sqlstm.sqadto[2] = (unsigned short )0;

            sqlstm.sqtdso[2] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=\

:b1";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1354;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from bank where BANK\_ID=:xbank; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from bank  where BANK\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1377;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tbill where BILL\_ID=:xbill; \*/

{

         struct sqlexd sqlstm;

         sqlstm.sqlvsn = 12;

         sqlstm.arrsiz = 12;

         sqlstm.sqladtp = &sqladt;

         sqlstm.sqltdsp = &sqltds;

         sqlstm.stmt = "delete  from tbill  where BILL\_ID=:b0";

         sqlstm.iters = (unsigned int  )1;

         sqlstm.offset = (unsigned int  )1396;

         sqlstm.cud = sqlcud0;

         sqlstm.sqlest = (unsigned char  \*)&sqlca;

         sqlstm.sqlety = (unsigned short)4352;

         sqlstm.occurs = (unsigned int  )0;

         sqlstm.sqhstv[0] = (         void  \*)&xbill;

         sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

         sqlstm.sqhsts[0] = (         int  )0;

         sqlstm.sqindv[0] = (         void  \*)0;

         sqlstm.sqinds[0] = (         int  )0;

         sqlstm.sqharm[0] = (unsigned int  )0;

         sqlstm.sqadto[0] = (unsigned short )0;

         sqlstm.sqtdso[0] = (unsigned short )0;

         sqlstm.sqphsv = sqlstm.sqhstv;

         sqlstm.sqphsl = sqlstm.sqhstl;

         sqlstm.sqphss = sqlstm.sqhsts;

         sqlstm.sqpind = sqlstm.sqindv;

         sqlstm.sqpins = sqlstm.sqinds;

         sqlstm.sqparm = sqlstm.sqharm;

         sqlstm.sqparc = sqlstm.sqharc;

         sqlstm.sqpadto = sqlstm.sqadto;

         sqlstm.sqptdso = sqlstm.sqtdso;

         sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tcal  where CALL\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1415;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xcall;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tphone where ACC\_NO=:cid AND PHONE\_NO=:cphone; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tphone  where (ACC\_NO=:b0 and PHONE\_\

NO=:b1)";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1434;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&cid;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)cphone;

            sqlstm.sqhstl[1] = (unsigned int  )12;

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql commit work; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1457;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tPHONE NUMBER DELETED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto q;

        }

        if(ck==3)

        {

            system("cls");

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have chosen to VIEW phone number";

            cout<<endl<<endl;

            cout<<"=============================================================================";

            cout<<"\n\n\n\n\n\n\n\n\n\t\t    Enter CUSTOMER id    : ";

            cin>>cid;

            selectphone(cid);

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto q;

        }

            if(ck==4)

        {

            system("cls");

            goto y;

        }

            if(ck==0||ck>4)

        {

            cout<<"Invalid choice..... Enter any Key";

            cin>>tem;

            goto q;

        }

      }

        case 3:

      {

        system("cls");

        int sel;

        z:

        system("cls");

        cout<<endl<<endl;

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

        cout<<endl;

        cout<<"\t\t\t\tBILLING  SECTION";

        cout<<endl;

        cout<<"\t\t\t\t================";

        cout<<endl<<endl<<endl;

        cout<<"\t\t\t\t1) ADD   \n\n";//input calulate

        cout<<"\t\t\t\t2) EDIT  \n\n";

        cout<<"\t\t\t\t3) DELETE\n\n";

        cout<<"\t\t\t\t4) SHOW  \n\n";//bill

        cout<<"\t\t\t\t5) BACK to MAIN";

        cout<<endl<<endl<<endl;

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

        cout<<endl<<"        ENTER YOUR CHOICE      : ";

        cin>>sel;

        if(sel==1)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To ADD BILL";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\t\tEnter CUSTOMER PHONE NUMBER to ADD BILL    : ";

            cin>>cphone;

            system("cls");

            cout<<"=======================================================================";

            insertbill(cphone,cid);

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tBILL ADDED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto z;

        }

        if(sel==2)

        {

            system("cls");

            cout<<"============================================================================="<<endl;

            cout<<"\t\t\tYou Have chosen to EDIT PHONE BILL";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\t\tEnter CUSTOMER PHONE NUMBER to ADD BILL    : ";

            cin>>cphone;

            editbill(cphone,cid);

            cout<<"============================================================================="<<endl;

            cout<<"\t\t\tCUSTOMER EDITED SUCCESSFULLY";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl<<"Enter any key....";

            cin>>r;

            goto z;

}

        if(sel==3)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To DELETE BILL";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\n\t\tEnter CUSTOMER PHONE NUMBER to DELETE BILL    : ";

            cin>>cphone;

            /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where ACC\_NO=:cid and PHONE\_NO=:cphone; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone\

 where (ACC\_NO=:b2 and PHONE\_NO=:b3)";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1472;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqhstv[2] = (         void  \*)&cid;

            sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[2] = (         int  )0;

            sqlstm.sqindv[2] = (         void  \*)0;

            sqlstm.sqinds[2] = (         int  )0;

            sqlstm.sqharm[2] = (unsigned int  )0;

            sqlstm.sqadto[2] = (unsigned short )0;

            sqlstm.sqtdso[2] = (unsigned short )0;

            sqlstm.sqhstv[3] = (         void  \*)cphone;

            sqlstm.sqhstl[3] = (unsigned int  )12;

            sqlstm.sqhsts[3] = (         int  )0;

            sqlstm.sqindv[3] = (         void  \*)0;

            sqlstm.sqinds[3] = (         int  )0;

            sqlstm.sqharm[3] = (unsigned int  )0;

            sqlstm.sqadto[3] = (unsigned short )0;

            sqlstm.sqtdso[3] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=\

:b1";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1503;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from bank where BANK\_ID=:xbank; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from bank  where BANK\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1526;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tbill where BILL\_ID=:xbill; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tbill  where BILL\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1545;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from tcal where CALL\_ID=:xcall; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from tcal  where CALL\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1564;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xcall;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql commit work; \*/

{

         struct sqlexd sqlstm;

         sqlstm.sqlvsn = 12;

         sqlstm.arrsiz = 12;

         sqlstm.sqladtp = &sqladt;

         sqlstm.sqltdsp = &sqltds;

         sqlstm.iters = (unsigned int  )1;

         sqlstm.offset = (unsigned int  )1583;

         sqlstm.cud = sqlcud0;

         sqlstm.sqlest = (unsigned char  \*)&sqlca;

         sqlstm.sqlety = (unsigned short)4352;

         sqlstm.occurs = (unsigned int  )0;

         sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tBILL DELETED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto z;

        }

        if (sel==4)

        {

            system("cls");

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have chosen to VIEW BILL";

            cout<<endl<<endl;

            cout<<"=============================================================================";

            cout<<"\n\n\n\n\n\n\n\n\n\t\t    Enter CUSTOMER id    : ";

            cin>>cid;

            cout<<"\t    Enter CUSTOMER PHONE NUMBER to VIEW BILL     : ";

            cin>>cphone;

            system("cls");

            selectbill(cphone,cid);

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto z;

        }

        if(sel==5)

        {

            system("cls");

            goto y;

        }

           if(sel==0||sel>5)

        {

            cout<<"Invalid choice..... Enter any Key";

            cin>>tem;

            goto z;

        }

      }

       case 4:

      {

        system("cls");

        int bk;

        d:

        system("cls");

        cout<<endl<<endl;

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

        cout<<endl;

        cout<<"\t\t\t\tBANK  DATABASE";

        cout<<endl;

        cout<<"\t\t\t\t=====================";

        cout<<endl<<endl<<endl;

        cout<<"\t\t\t\t1) ADD   \n\n";//input calulate

        cout<<"\t\t\t\t2) EDIT  \n\n";

        cout<<"\t\t\t\t3) DELETE\n\n";

        cout<<"\t\t\t\t4) SHOW  \n\n";//bill

        cout<<"\t\t\t\t5) BACK to MAIN";

        cout<<endl<<endl<<endl;

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

        cout<<endl<<"        ENTER YOUR CHOICE      : ";

        cin>>bk;

         if(bk==1)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To ADD BANK DETAILS";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\t\tEnter CUSTOMER PHONE NUMBER to ADD BANK DETAILS    : ";

            cin>>cphone;

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

            insertbank(cphone,cid);

            cin.get();

            system("cls");

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tBANK DETAILS ADDED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto d;

        }

         if(bk==2)

        {

            system("cls");

            cout<<"============================================================================="<<endl;

            cout<<"\t\t\tYou Have chosen to BANK DETAILS";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\t\tEnter CUSTOMER PHONE NUMBER to ADD BANK DETAILS    : ";

            cin>>cphone;

            editbank(cphone,cid);

            cout<<"============================================================================="<<endl;

            cout<<"\t\t\tBANK DETAILS EDITED SUCCESSFULLY";

            cout<<endl;

            cout<<"=============================================================================";

            cout<<endl<<"Enter any key....";

            cin>>r;

            goto d;

        }

         if(bk==3)

        {

            system("cls");

            cout<<"======================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have Chosen To BANK DETAILS";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<"\n\n\t\tEnter CUSTOMER id   : ";

            cin>>cid;

            cout<<"\n\t\tEnter CUSTOMER PHONE NUMBER to DELETE BANK DETAILS    : ";

            cin>>cphone;

            /\* exec sql select bill\_id,call\_id into :xbill,:xcall from tphone where ACC\_NO=:cid and PHONE\_NO=:cphone; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "select bill\_id ,call\_id into :b0,:b1  from tphone\

 where (ACC\_NO=:b2 and PHONE\_NO=:b3)";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1598;

            sqlstm.selerr = (unsigned short)1;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbill;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqhstv[1] = (         void  \*)&xcall;

            sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[1] = (         int  )0;

            sqlstm.sqindv[1] = (         void  \*)0;

            sqlstm.sqinds[1] = (         int  )0;

            sqlstm.sqharm[1] = (unsigned int  )0;

            sqlstm.sqadto[1] = (unsigned short )0;

            sqlstm.sqtdso[1] = (unsigned short )0;

            sqlstm.sqhstv[2] = (         void  \*)&cid;

            sqlstm.sqhstl[2] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[2] = (         int  )0;

            sqlstm.sqindv[2] = (         void  \*)0;

            sqlstm.sqinds[2] = (         int  )0;

            sqlstm.sqharm[2] = (unsigned int  )0;

            sqlstm.sqadto[2] = (unsigned short )0;

            sqlstm.sqtdso[2] = (unsigned short )0;

            sqlstm.sqhstv[3] = (         void  \*)cphone;

            sqlstm.sqhstl[3] = (unsigned int  )12;

            sqlstm.sqhsts[3] = (         int  )0;

            sqlstm.sqindv[3] = (         void  \*)0;

            sqlstm.sqinds[3] = (         int  )0;

            sqlstm.sqharm[3] = (unsigned int  )0;

            sqlstm.sqadto[3] = (unsigned short )0;

            sqlstm.sqtdso[3] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql select BANK\_ID into :xbank from tcal where CALL\_ID=:xcall; \*/

{

         struct sqlexd sqlstm;

         sqlstm.sqlvsn = 12;

         sqlstm.arrsiz = 12;

         sqlstm.sqladtp = &sqladt;

         sqlstm.sqltdsp = &sqltds;

         sqlstm.stmt = "select BANK\_ID into :b0  from tcal where CALL\_ID=:b1";

         sqlstm.iters = (unsigned int  )1;

         sqlstm.offset = (unsigned int  )1629;

         sqlstm.selerr = (unsigned short)1;

         sqlstm.cud = sqlcud0;

         sqlstm.sqlest = (unsigned char  \*)&sqlca;

         sqlstm.sqlety = (unsigned short)4352;

         sqlstm.occurs = (unsigned int  )0;

         sqlstm.sqhstv[0] = (         void  \*)&xbank;

         sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

         sqlstm.sqhsts[0] = (         int  )0;

         sqlstm.sqindv[0] = (         void  \*)0;

         sqlstm.sqinds[0] = (         int  )0;

         sqlstm.sqharm[0] = (unsigned int  )0;

         sqlstm.sqadto[0] = (unsigned short )0;

         sqlstm.sqtdso[0] = (unsigned short )0;

         sqlstm.sqhstv[1] = (         void  \*)&xcall;

         sqlstm.sqhstl[1] = (unsigned int  )sizeof(int);

         sqlstm.sqhsts[1] = (         int  )0;

         sqlstm.sqindv[1] = (         void  \*)0;

         sqlstm.sqinds[1] = (         int  )0;

         sqlstm.sqharm[1] = (unsigned int  )0;

         sqlstm.sqadto[1] = (unsigned short )0;

         sqlstm.sqtdso[1] = (unsigned short )0;

         sqlstm.sqphsv = sqlstm.sqhstv;

         sqlstm.sqphsl = sqlstm.sqhstl;

         sqlstm.sqphss = sqlstm.sqhsts;

         sqlstm.sqpind = sqlstm.sqindv;

         sqlstm.sqpins = sqlstm.sqinds;

         sqlstm.sqparm = sqlstm.sqharm;

         sqlstm.sqparc = sqlstm.sqharc;

         sqlstm.sqpadto = sqlstm.sqadto;

         sqlstm.sqptdso = sqlstm.sqtdso;

         sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql delete from bank where BANK\_ID=:xbank; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.stmt = "delete  from bank  where BANK\_ID=:b0";

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1652;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlstm.sqhstv[0] = (         void  \*)&xbank;

            sqlstm.sqhstl[0] = (unsigned int  )sizeof(int);

            sqlstm.sqhsts[0] = (         int  )0;

            sqlstm.sqindv[0] = (         void  \*)0;

            sqlstm.sqinds[0] = (         int  )0;

            sqlstm.sqharm[0] = (unsigned int  )0;

            sqlstm.sqadto[0] = (unsigned short )0;

            sqlstm.sqtdso[0] = (unsigned short )0;

            sqlstm.sqphsv = sqlstm.sqhstv;

            sqlstm.sqphsl = sqlstm.sqhstl;

            sqlstm.sqphss = sqlstm.sqhsts;

            sqlstm.sqpind = sqlstm.sqindv;

            sqlstm.sqpins = sqlstm.sqinds;

            sqlstm.sqparm = sqlstm.sqharm;

            sqlstm.sqparc = sqlstm.sqharc;

            sqlstm.sqpadto = sqlstm.sqadto;

            sqlstm.sqptdso = sqlstm.sqtdso;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            /\* exec sql commit work; \*/

{

            struct sqlexd sqlstm;

            sqlstm.sqlvsn = 12;

            sqlstm.arrsiz = 12;

            sqlstm.sqladtp = &sqladt;

            sqlstm.sqltdsp = &sqltds;

            sqlstm.iters = (unsigned int  )1;

            sqlstm.offset = (unsigned int  )1671;

            sqlstm.cud = sqlcud0;

            sqlstm.sqlest = (unsigned char  \*)&sqlca;

            sqlstm.sqlety = (unsigned short)4352;

            sqlstm.occurs = (unsigned int  )0;

            sqlcxt((void \*\*)0, &sqlctx, &sqlstm, &sqlfpn);

}

            cout<<"=======================================================================";

            cout<<endl;

            cout<<"\t\t\tBANK DETAILS  DELETED SUCCESSFULLY";

            cout<<endl<<endl;

            cout<<"=======================================================================";

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto d;

        }

         if(bk==4)

        {

            system("cls");

            cout<<"=============================================================================";

            cout<<endl;

            cout<<"\t\t\tYou Have chosen to VIEW BANK DETAILS";

            cout<<endl<<endl;

            cout<<"=============================================================================";

            cout<<"\n\n\n\n\n\n\n\n\n\t\t    Enter CUSTOMER id    : ";

            cin>>cid;

            cout<<"\t    Enter CUSTOMER PHONE NUMBER to VIEW BANK DETAILS     : ";

            cin>>cphone;

            selectbank(cphone,cid);

            cout<<endl<<"Enter any key...";

            cin>>r;

            goto d;

        }

         if(bk==5)

        {

            system("cls");

            goto y;

        }

         if(bk==0||bk>5)

        {

            cout<<"\t\t\tInvalid choice..... Enter any Key";

            cin>>tem;

            goto d;

        }

    }

        case 5:

      {

        char f[1];

        system("cls");

        cout<<"=============================================================================";

        cout<<endl;

        cout<<"\t\t\t\tBILLING DETAILS";

        cout<<endl<<endl;

        cout<<"=============================================================================";

        cout<<endl;

        cout<<"\t\t STD RATE = Rs.1/minute"<<endl<<endl;

        cout<<"\t\t National Roaming Outgoing RATE = Rs.2/minute"<<endl<<endl;

        cout<<"\t\t National Roaming Incoming RATE = Rs.2.5/minute"<<endl<<endl;

        cout<<"\t\t SMS RATE = Rs.0.50/SMS"<<endl<<endl;

        cout<<"\t\t MMS RATE = Rs.3/MMS"<<endl<<endl;

        cout<<"\t\t GPRS RATE = Rs.0.10/kb"<<endl<<endl;

        cout<<"\t\t ISD RATE = Rs.8/minute"<<endl<<endl;

        cout<<"\t\t International Roaming Outgoing RATE = Rs.15/minute"<<endl<<endl;

        cout<<"\t\t International Roaming Incoming RATE = Rs.17/minute"<<endl<<endl;

        cin>>f;

        system("cls");

        goto y;

      }

        case 6:

      {

        system("cls");

        char s;

        cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\t\t\tAre You Sure ??? ( Enter 'y' for YES and ANY key for NO )    :";

        cin>>s;

        if(s=='y')

        exit(0);

        if(s!='y')

         {

            system("cls");

             goto y;

        }

        default:

        system("cls");

        cout<<"\t\t\t=============================================================================";

        cout<<endl<<"\t\t\tINVALID CHOICE TRY AGAIN   ";

        cout<<endl;

        cout<<"\t\t\t=============================================================================";

        cout<<endl<<"\t\tEnter any key...";

        cin>>r;

        goto y;

      }

    }

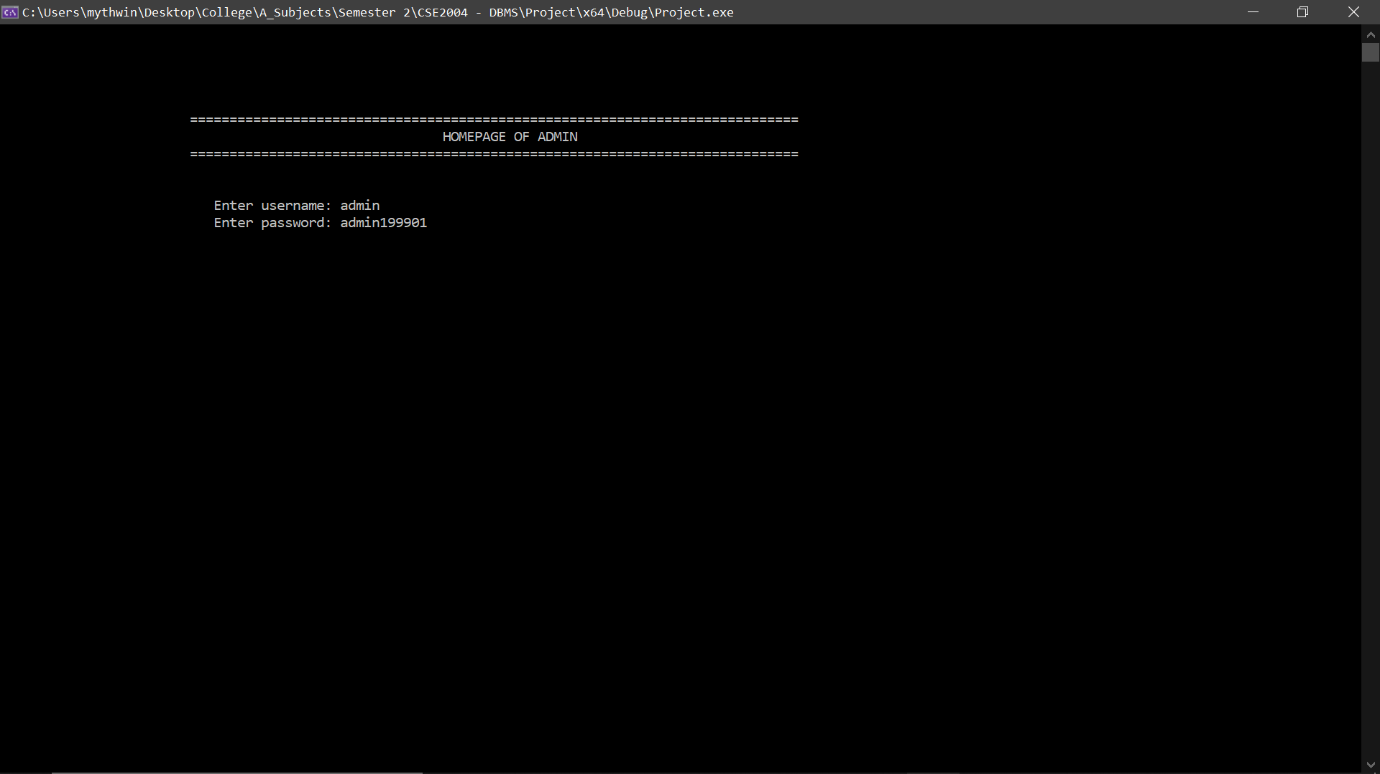
    }

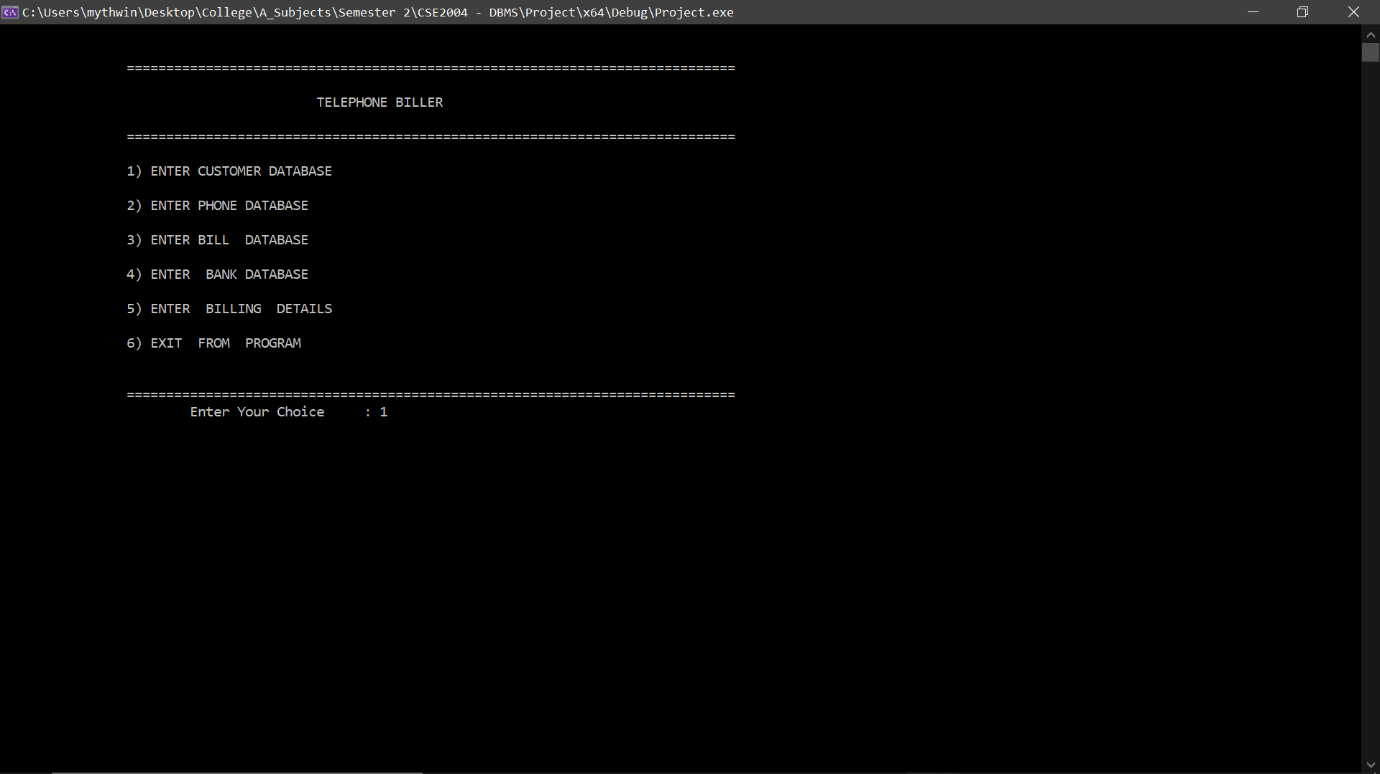
    else

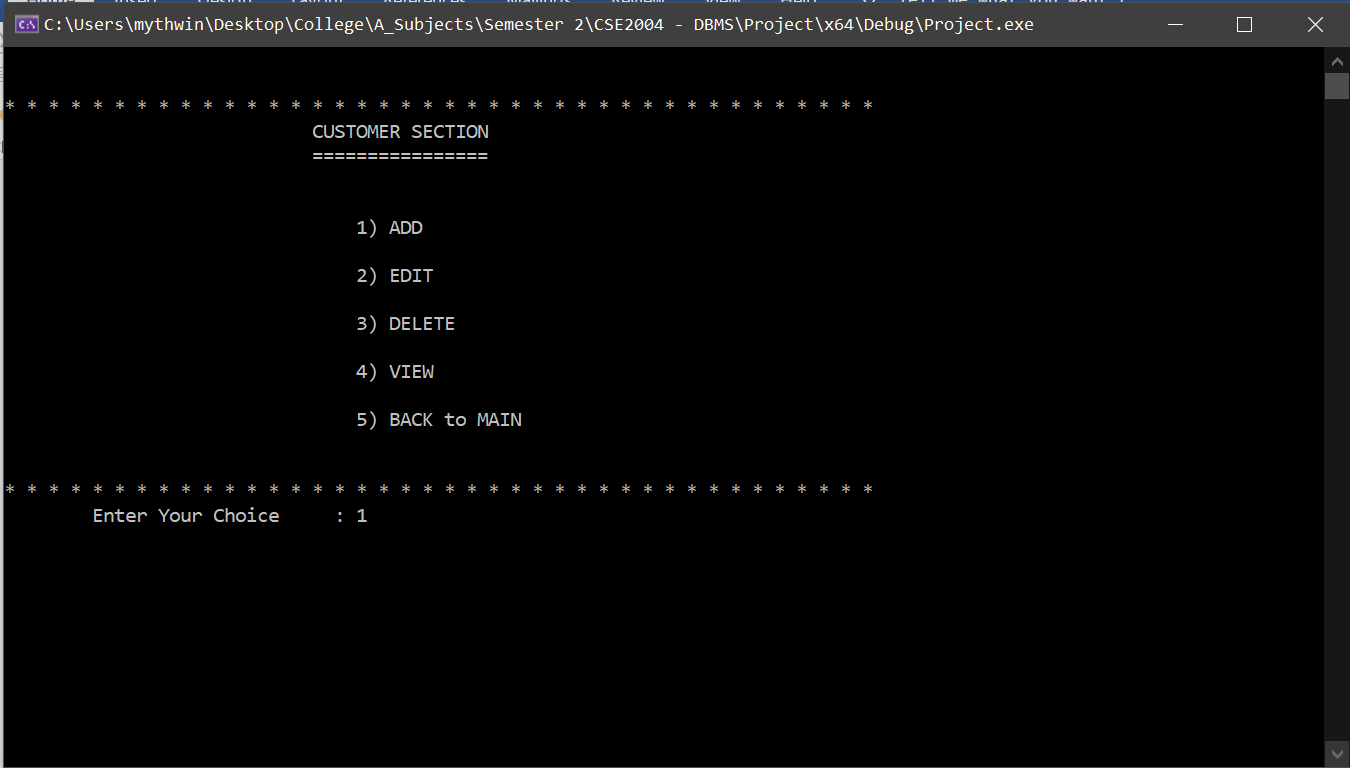
    cout<<"Error code: "<<sqlca.sqlcode<<endl<<"Message: "<<sqlca.sqlerrm.sqlerrmc;

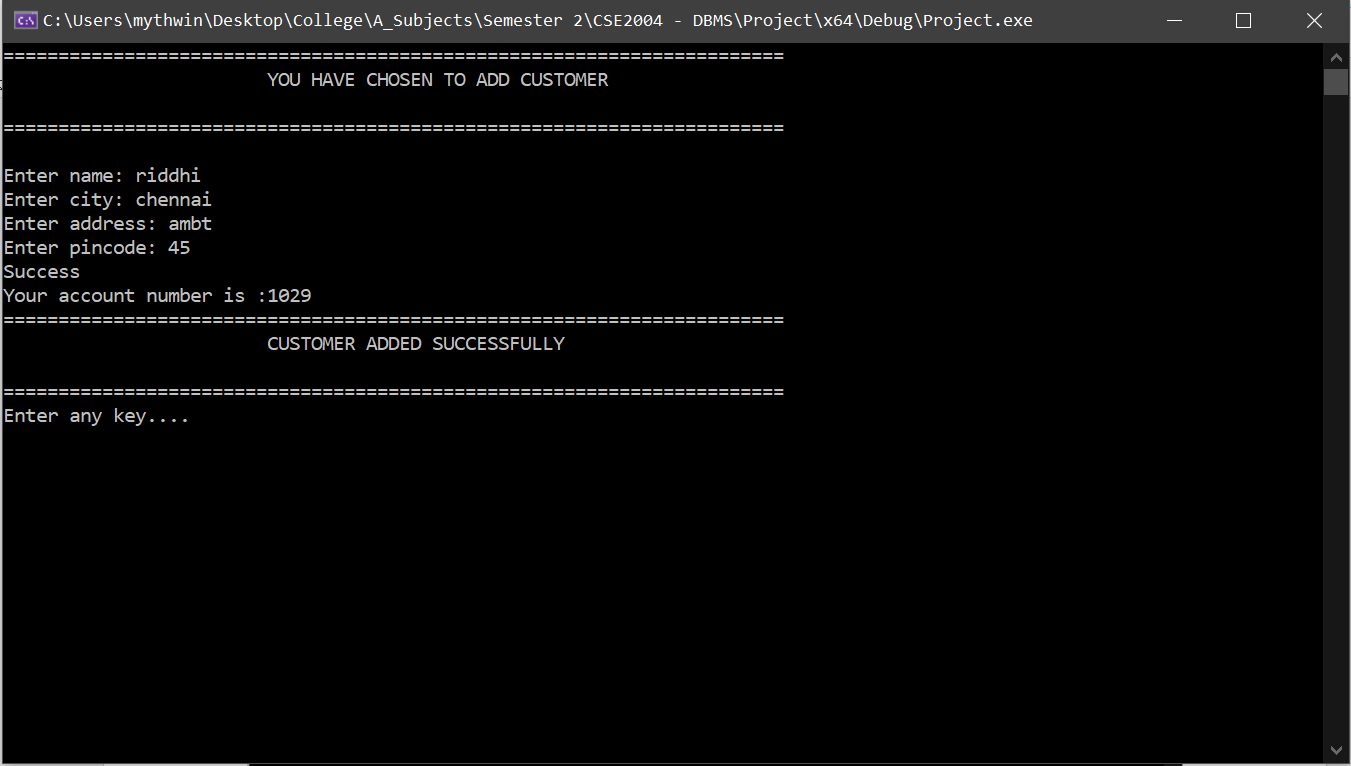
    return 0;}

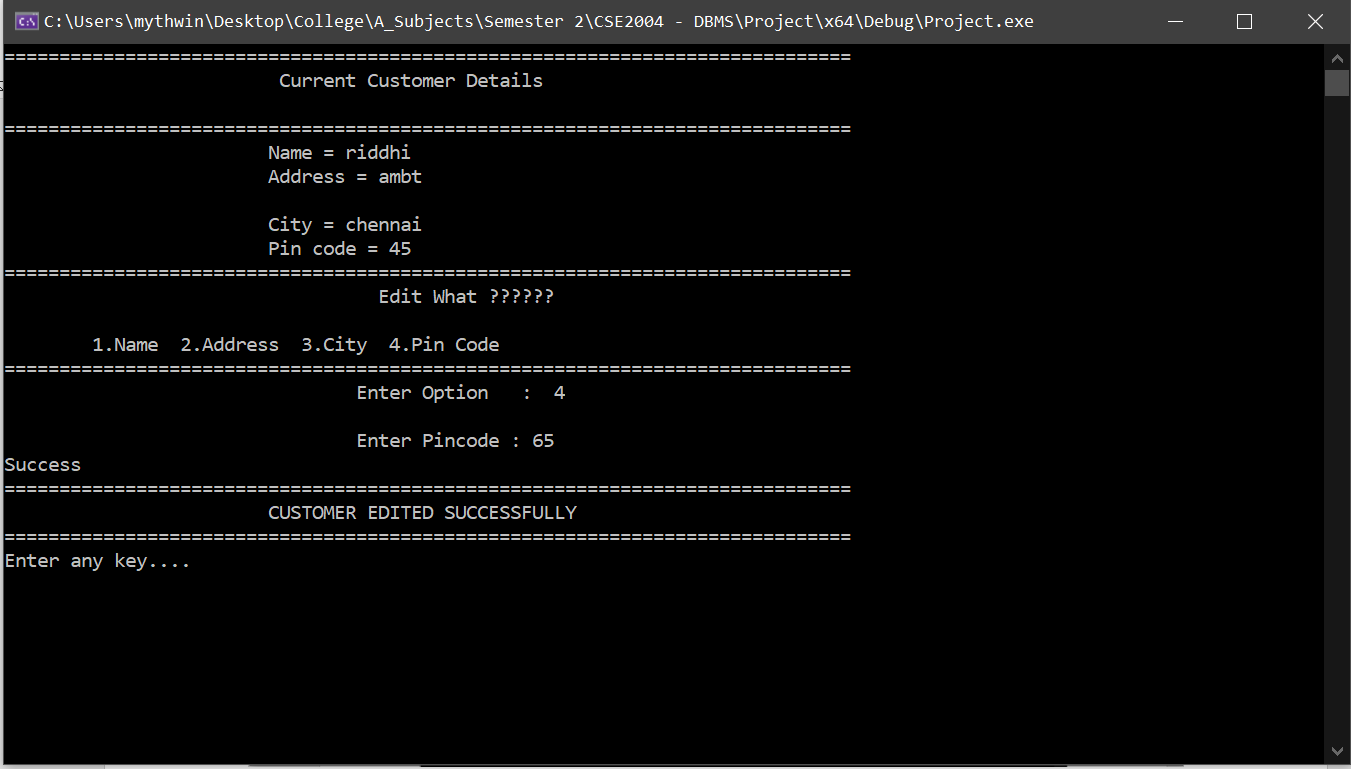
* **Output:**

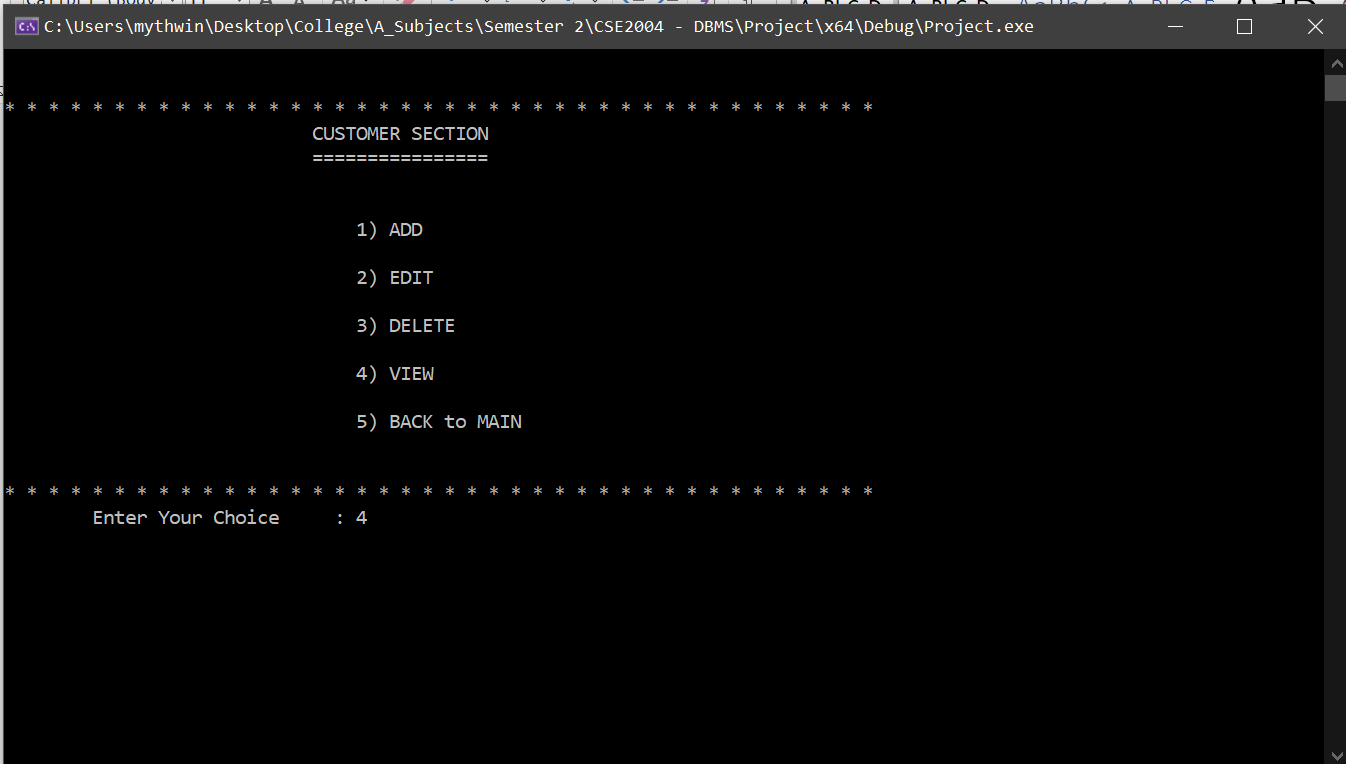


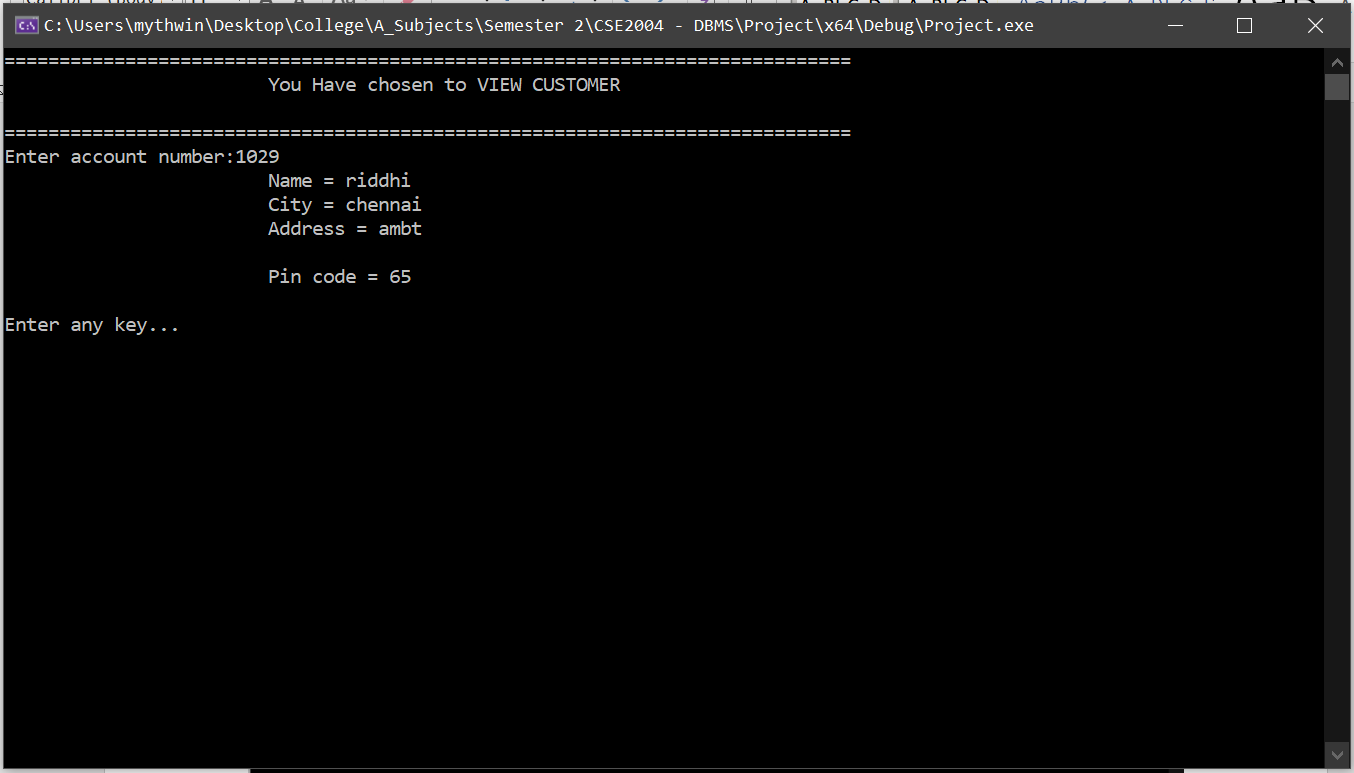


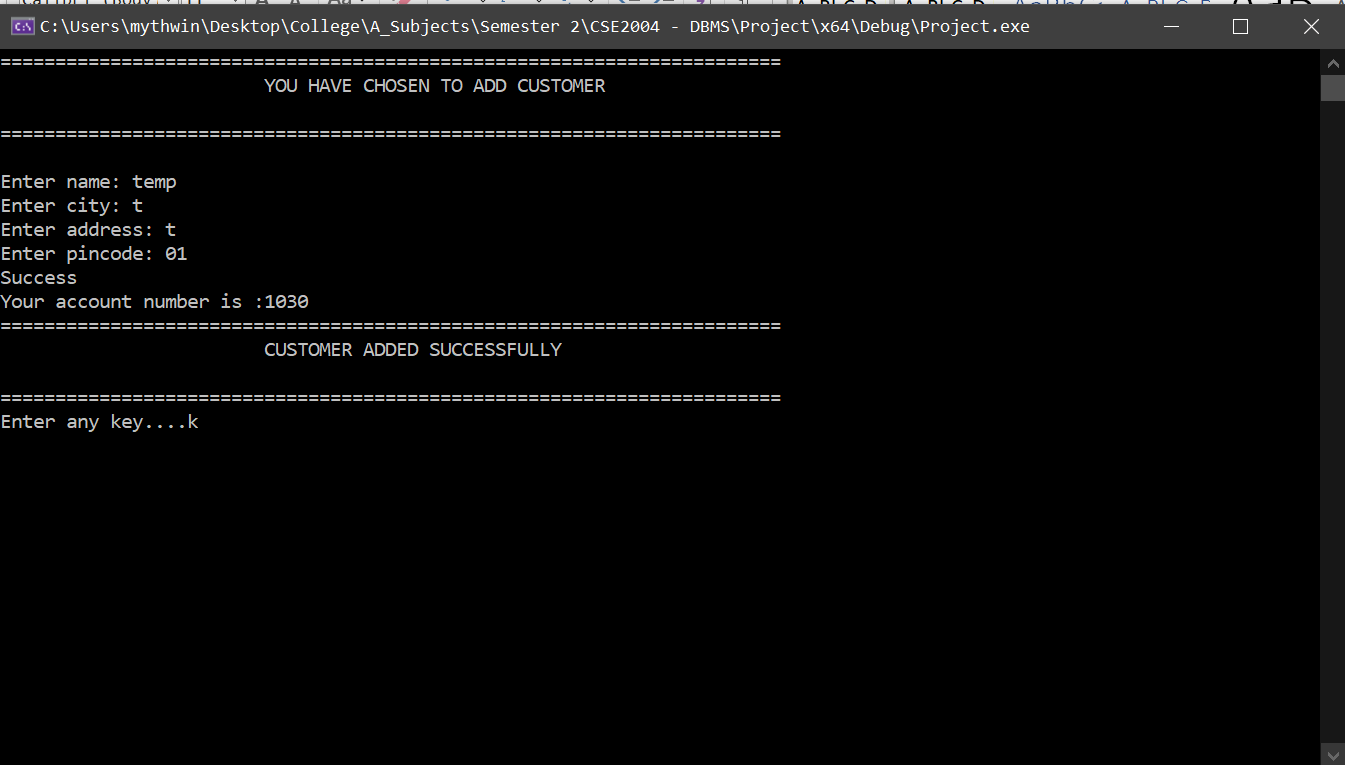


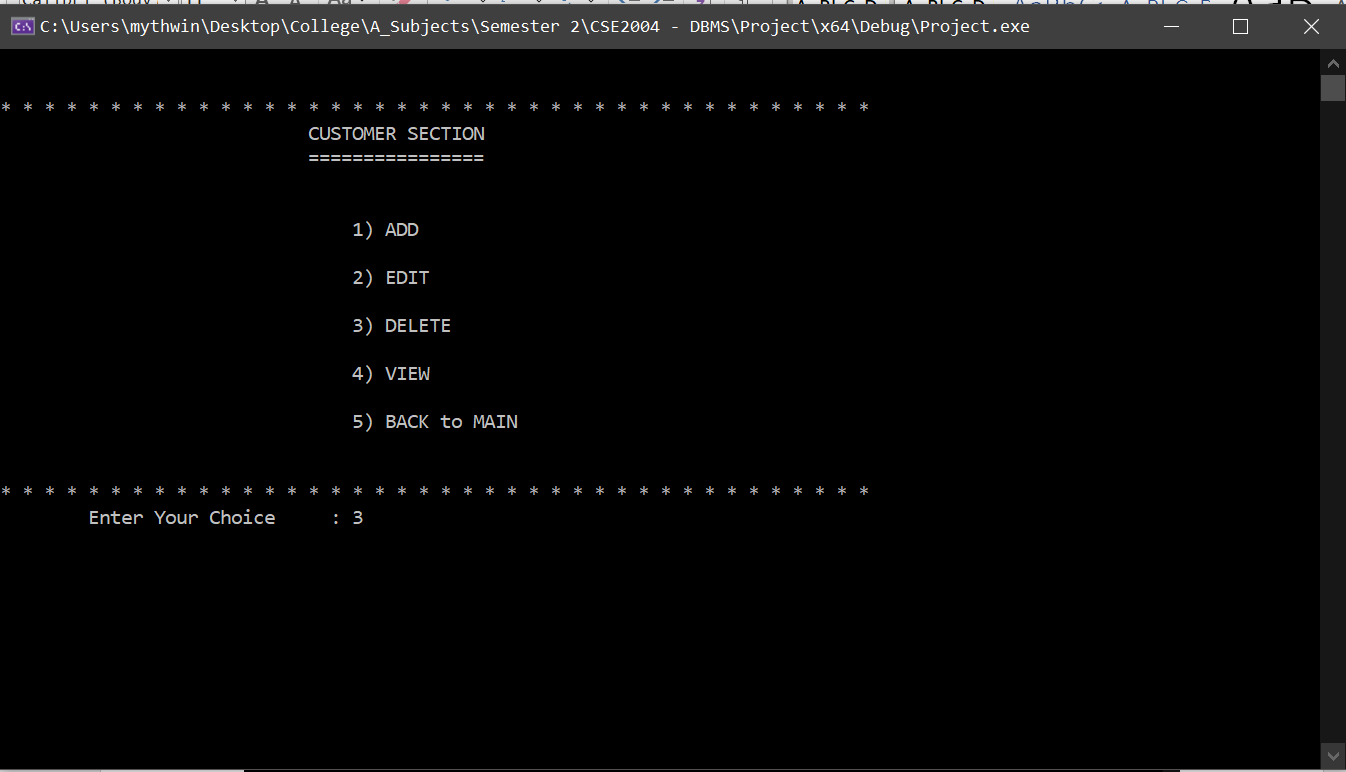


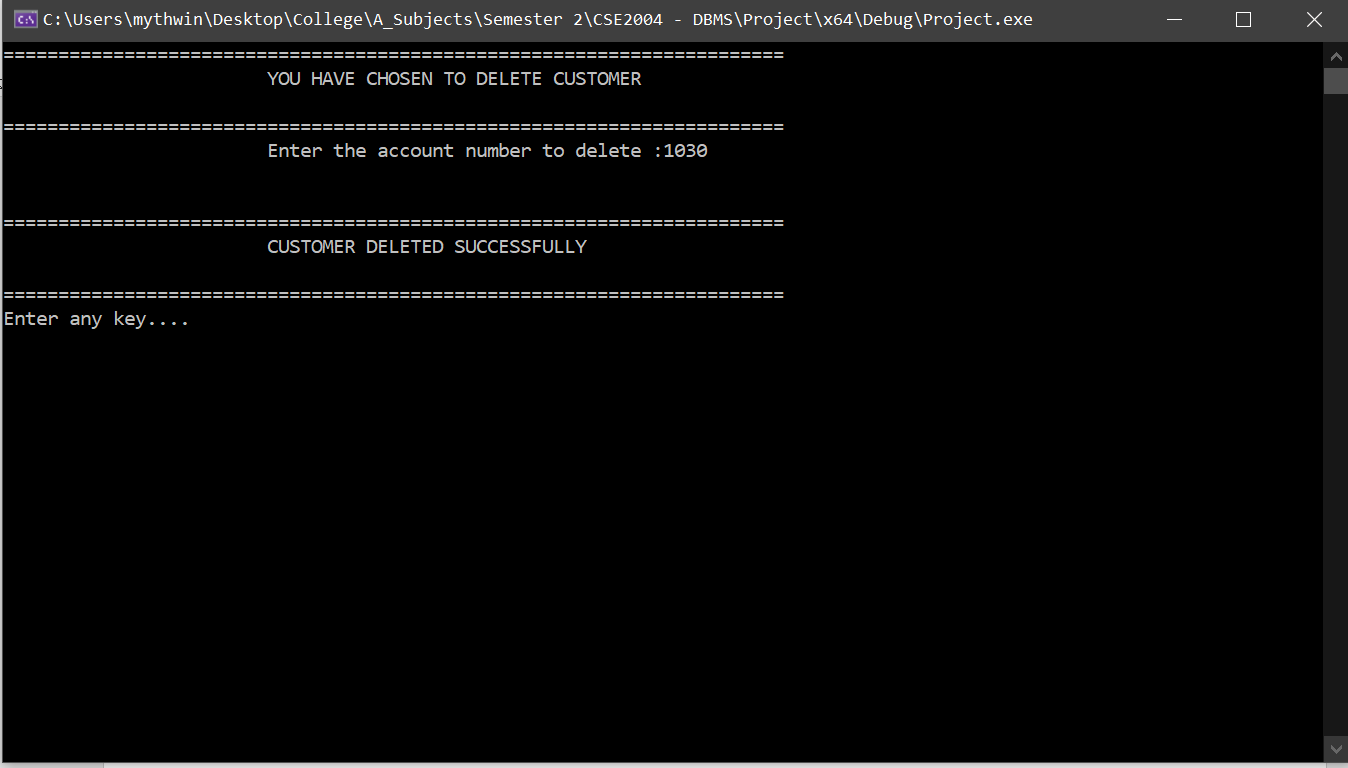


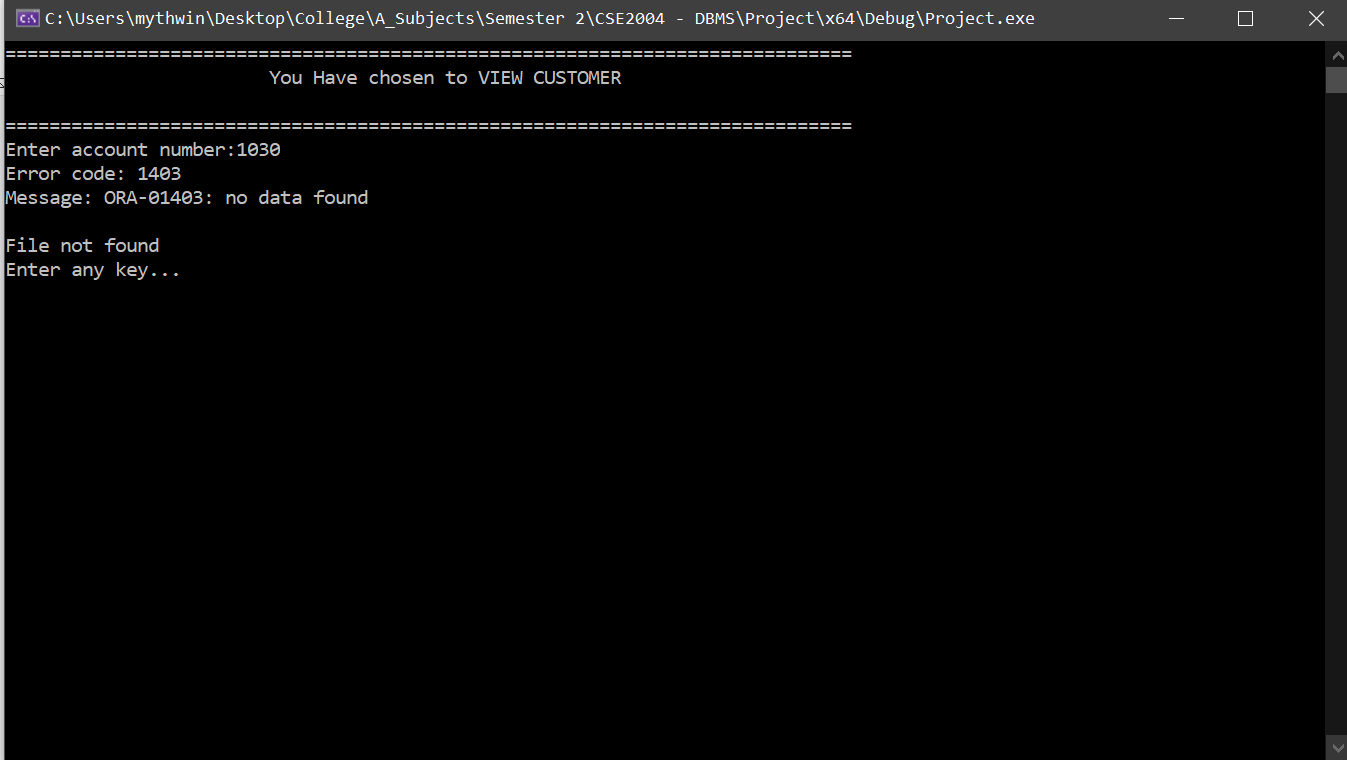


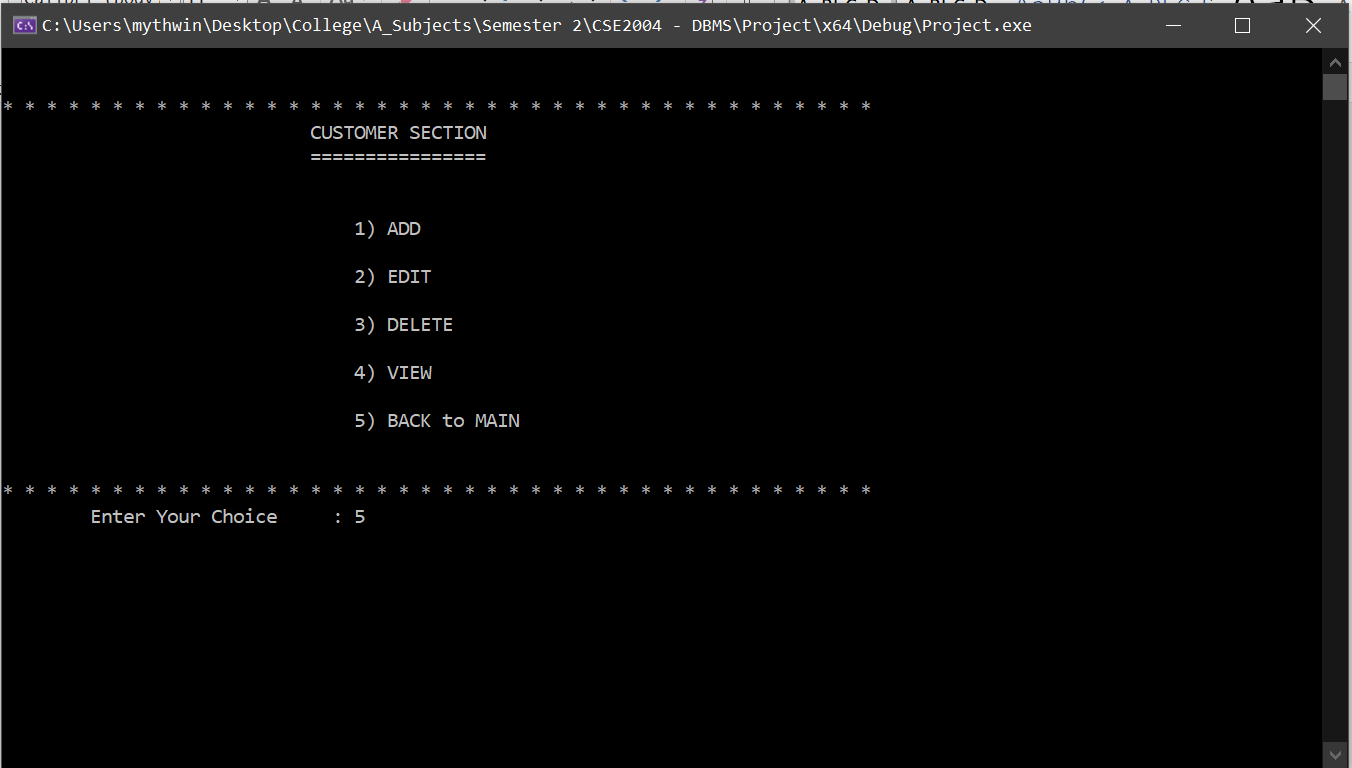


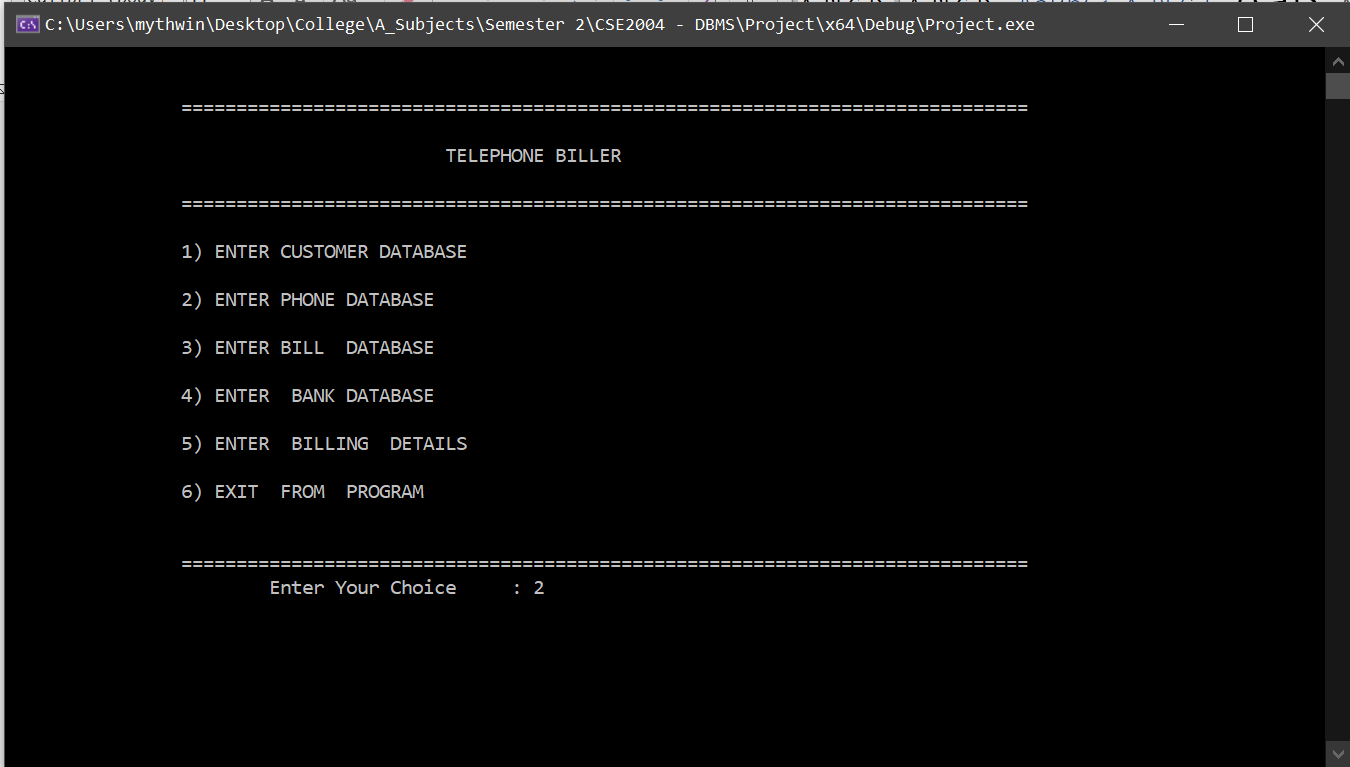


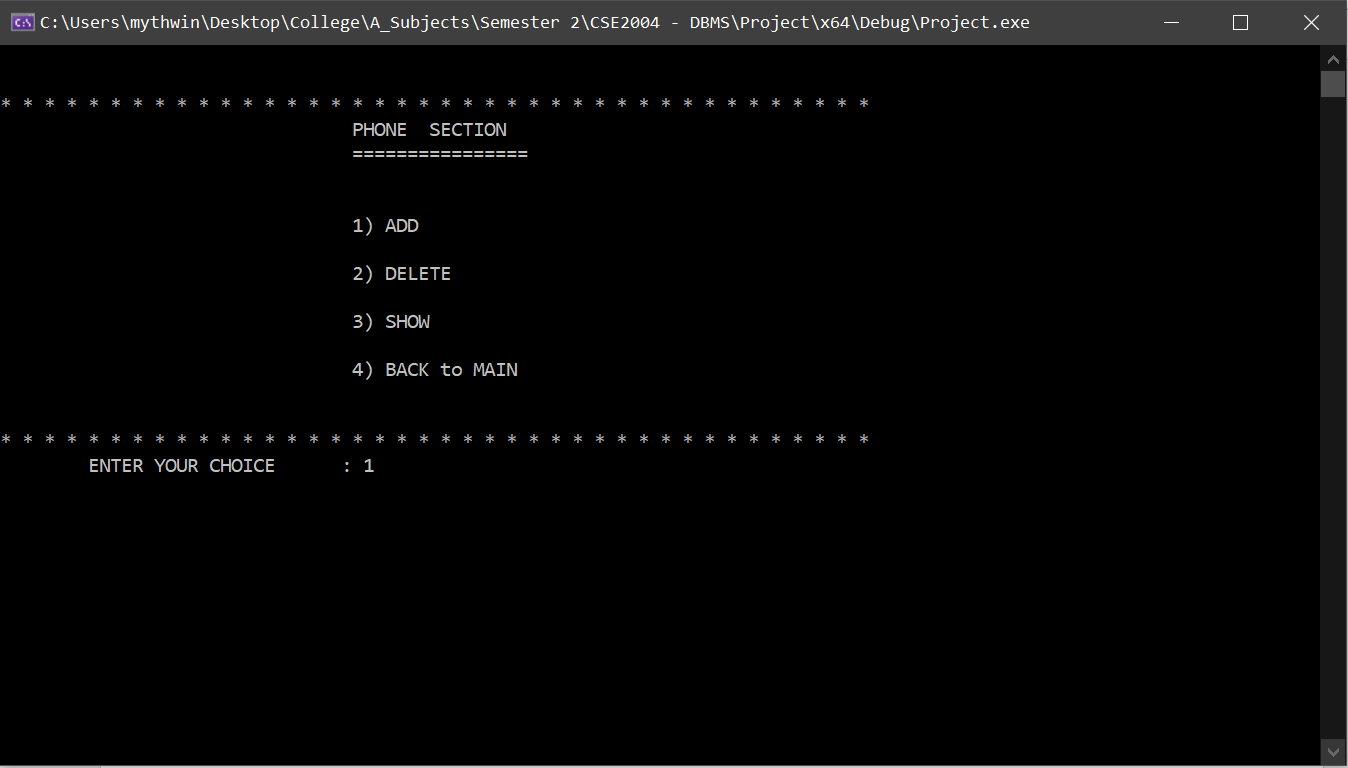


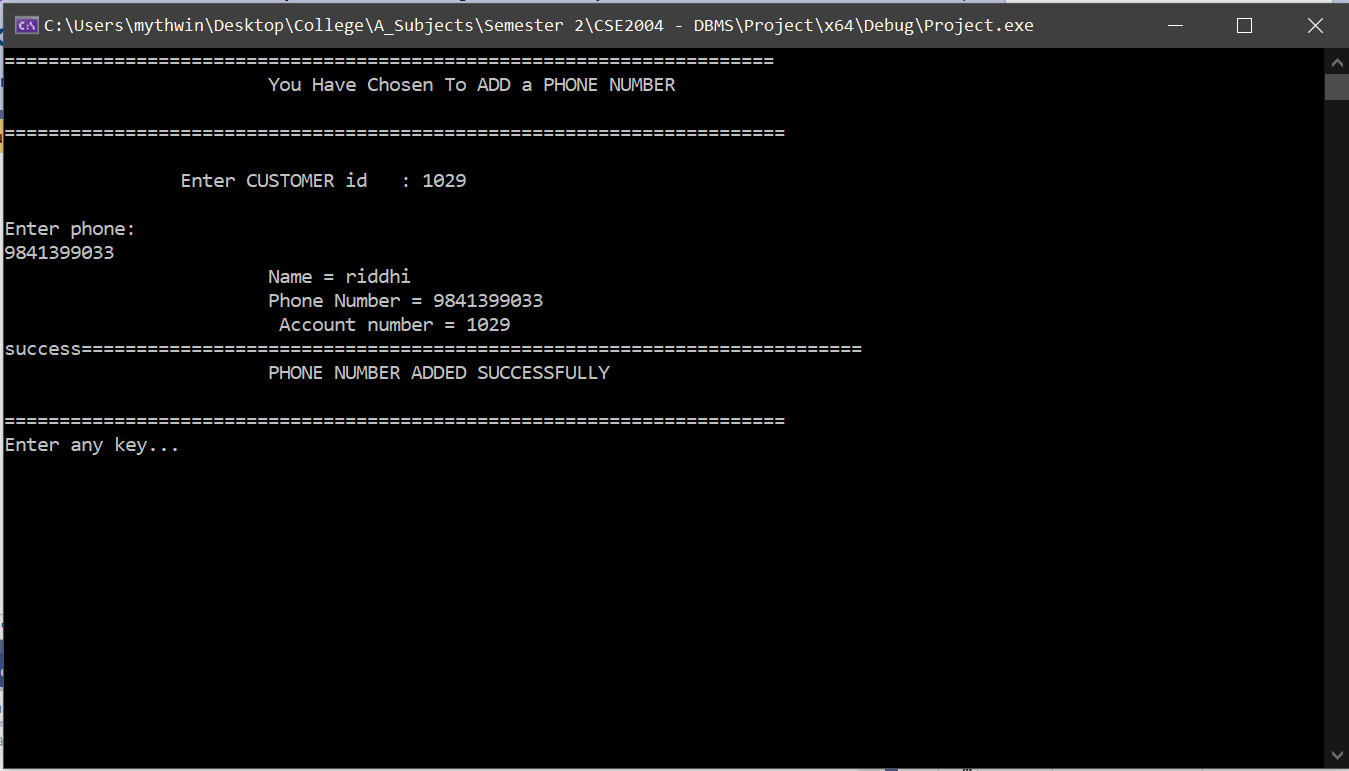


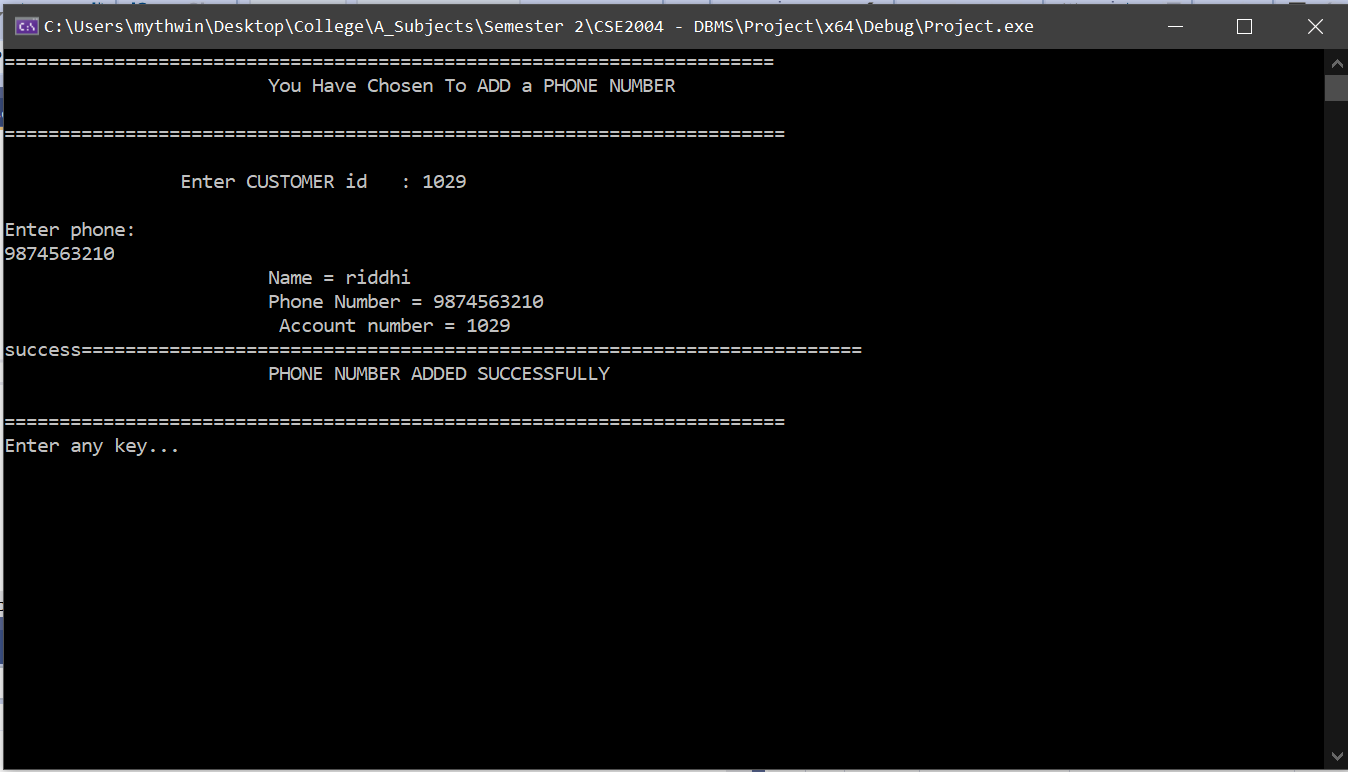


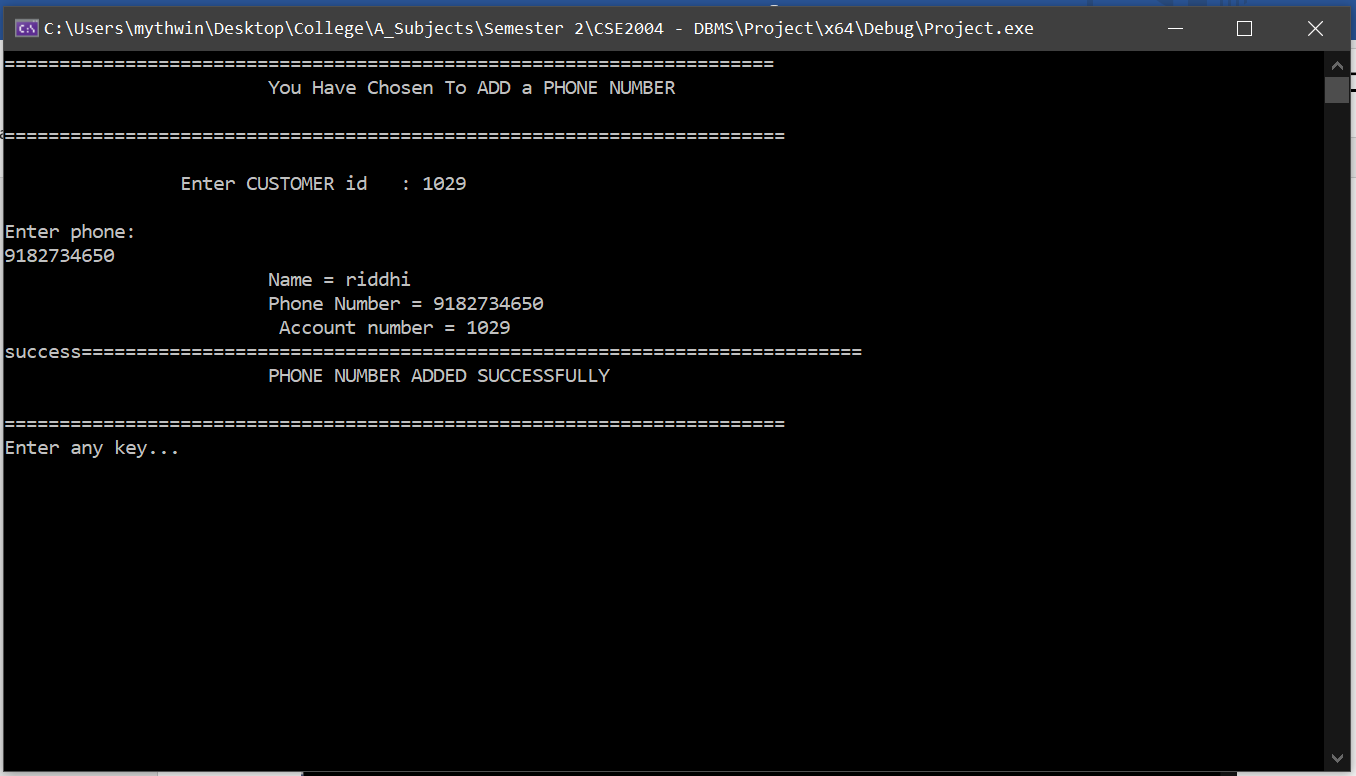


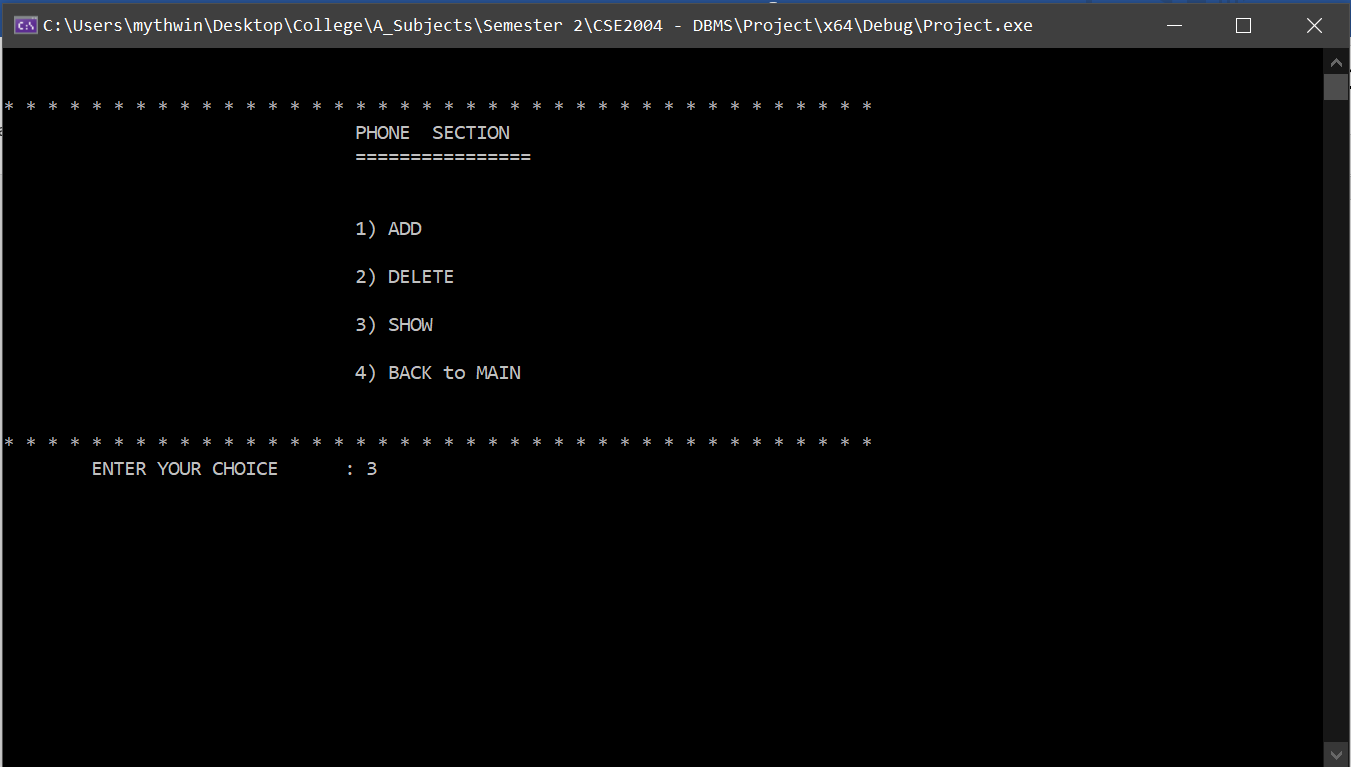


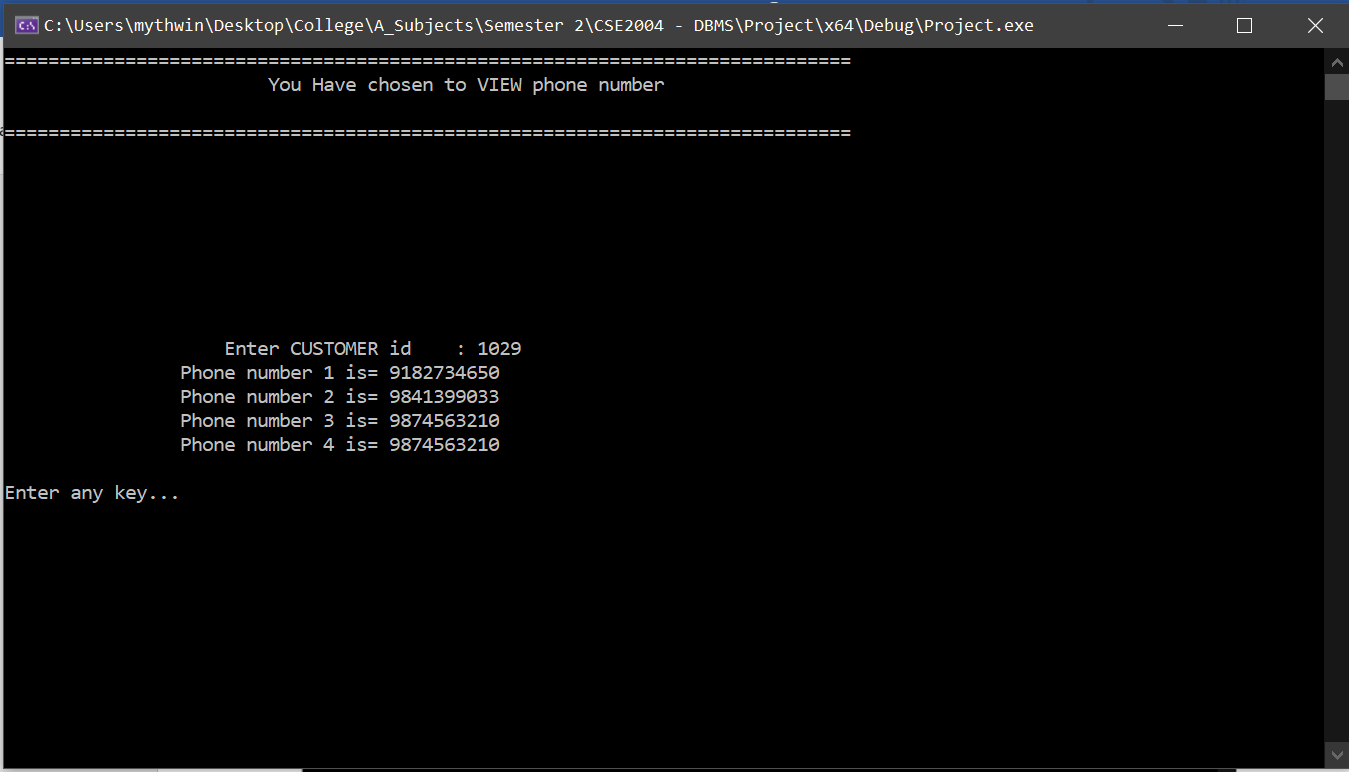


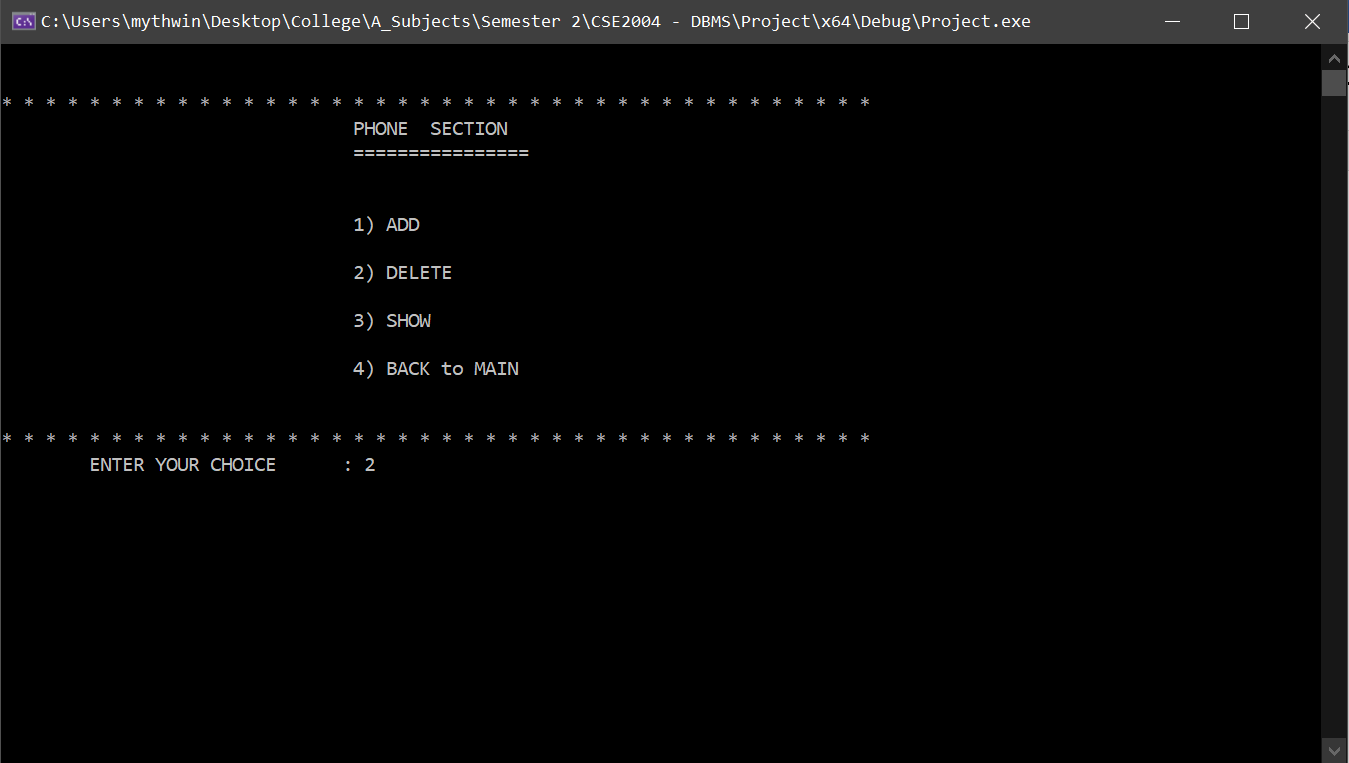


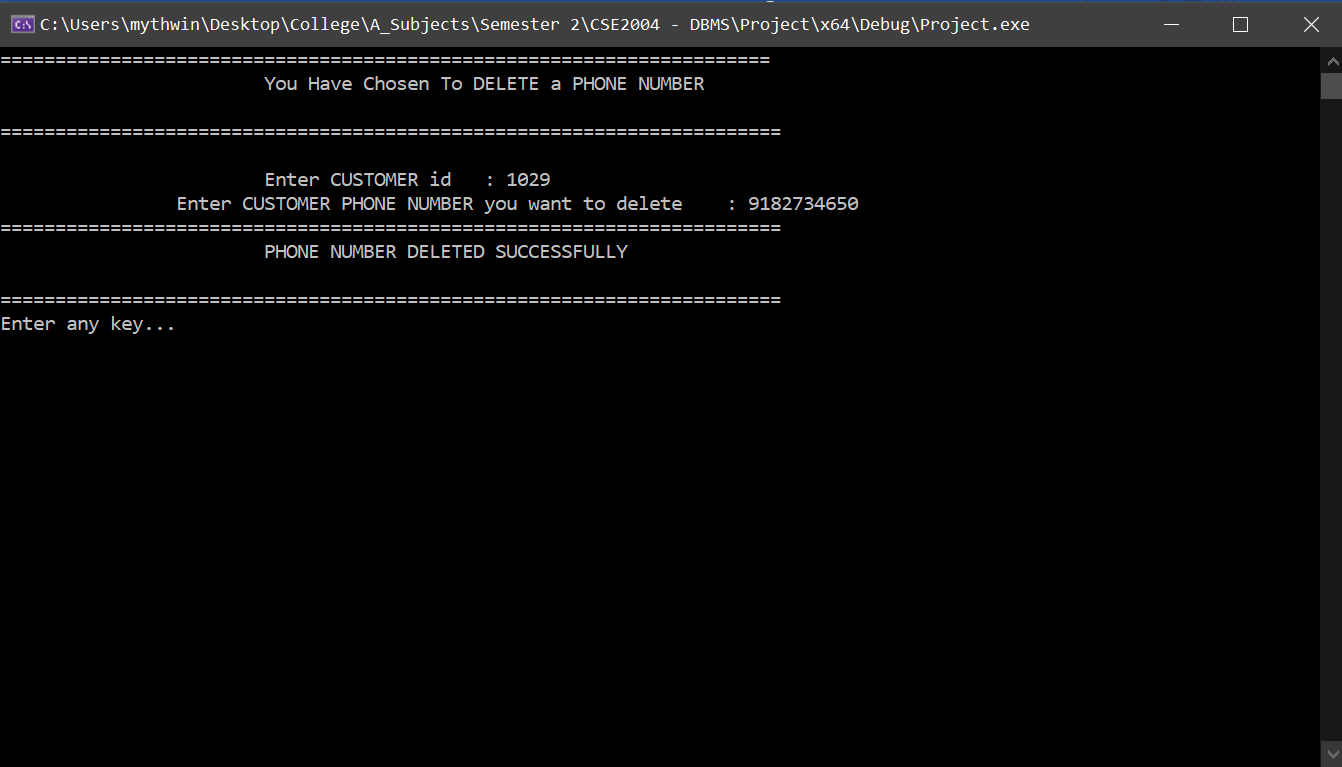


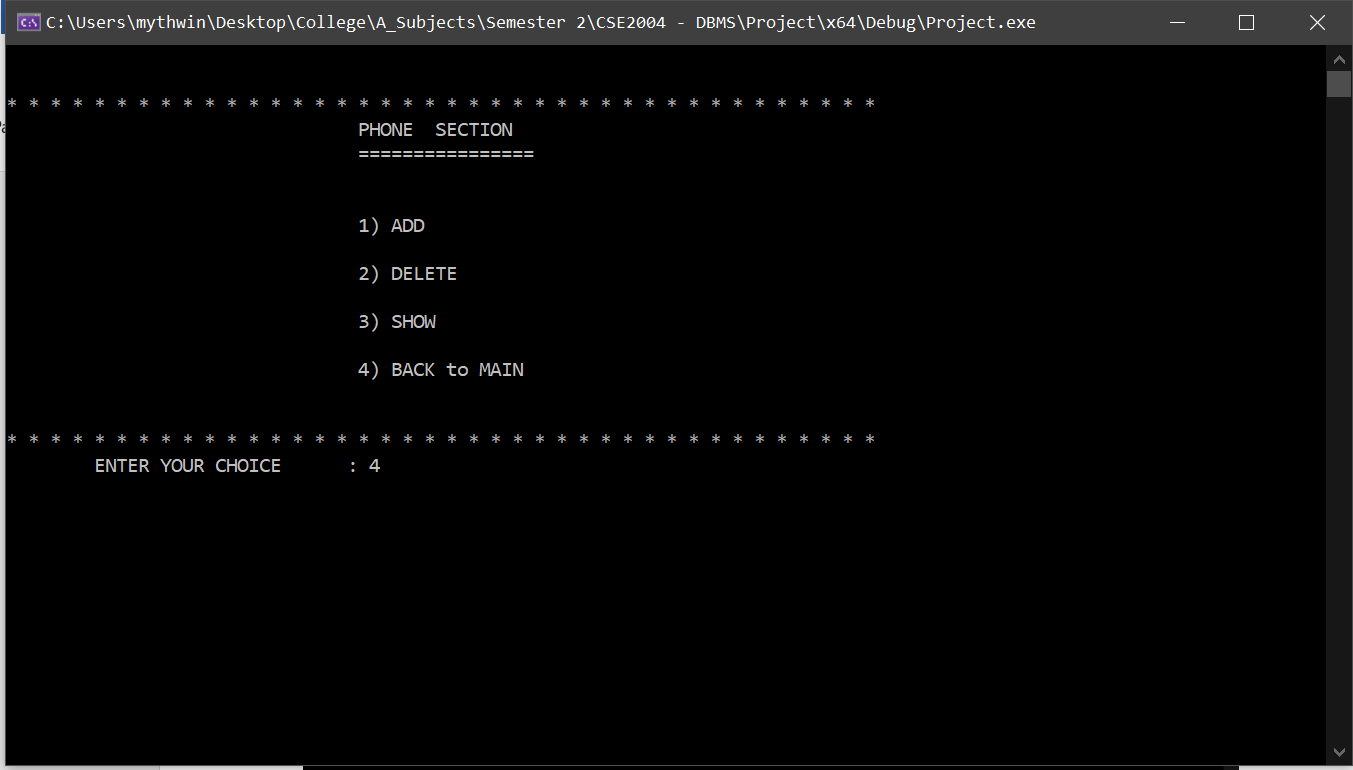


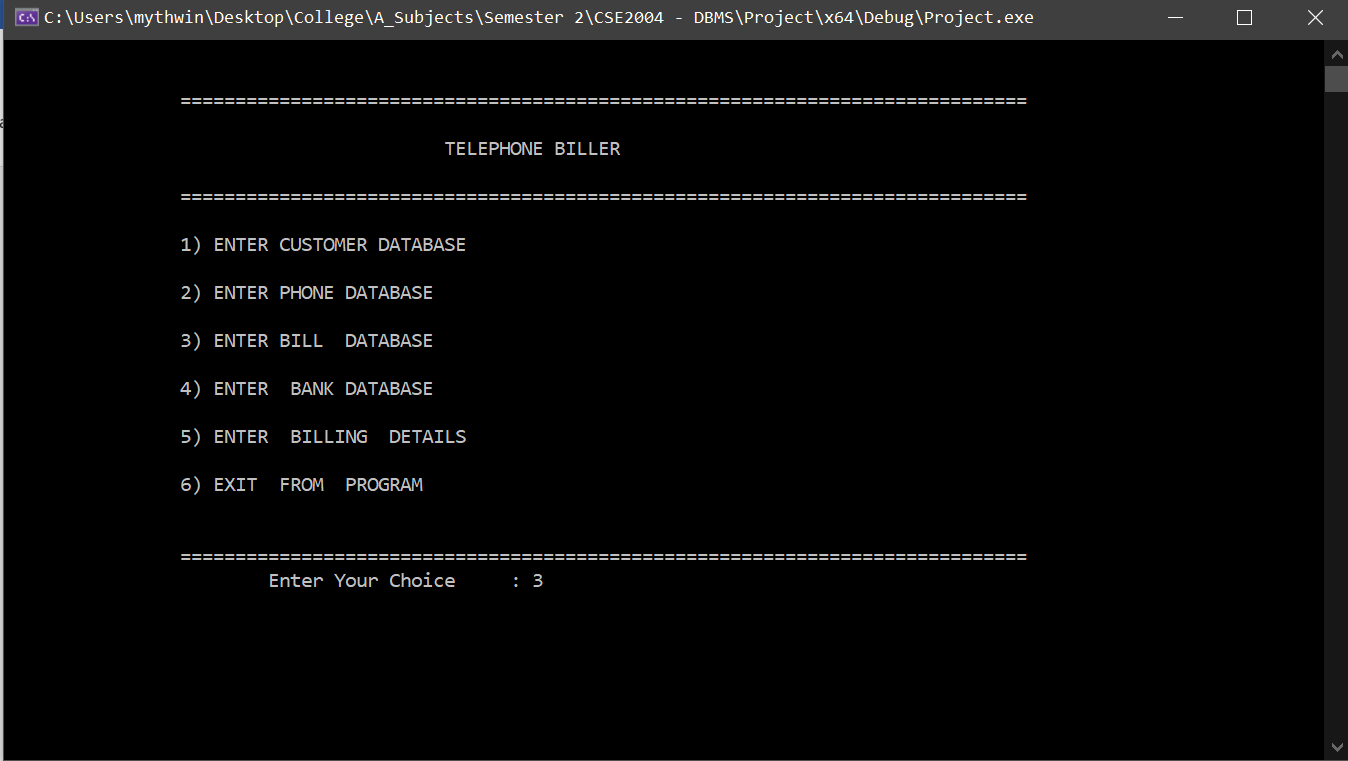


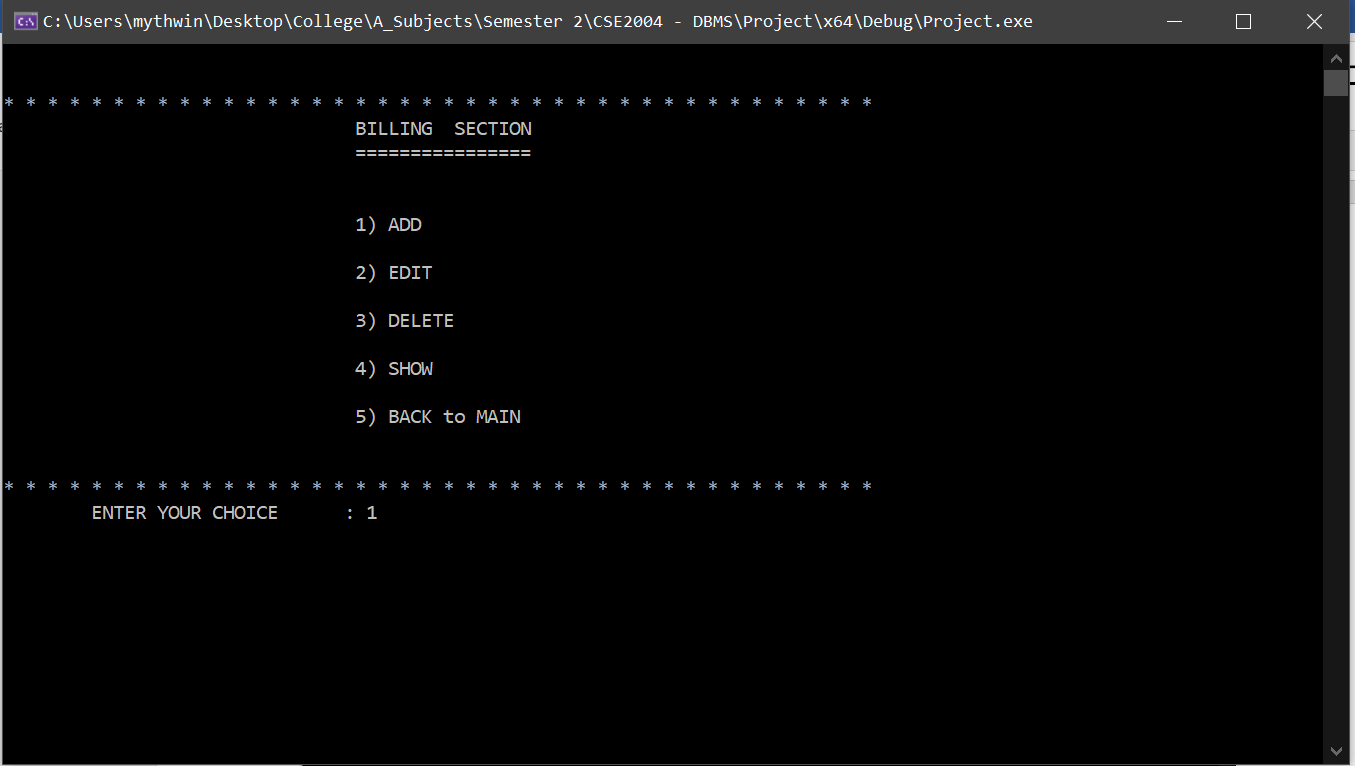


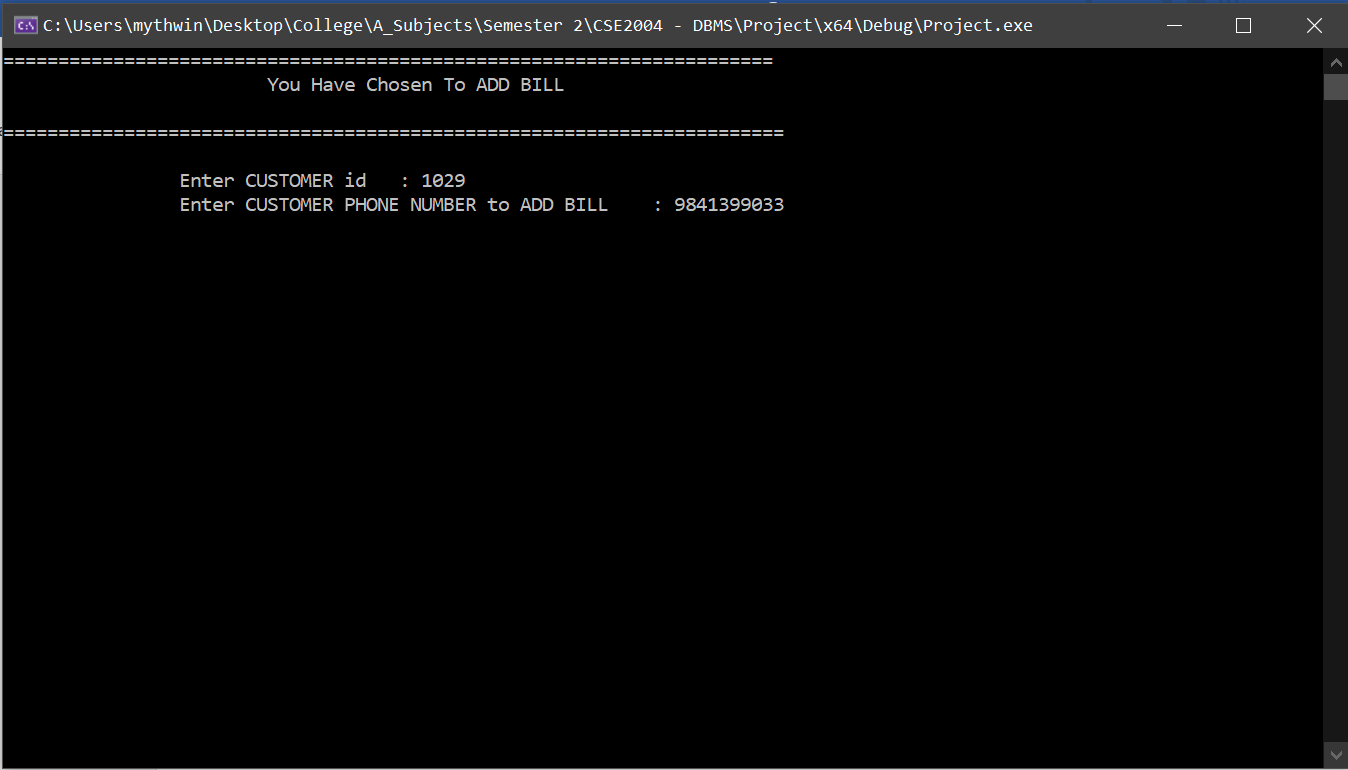


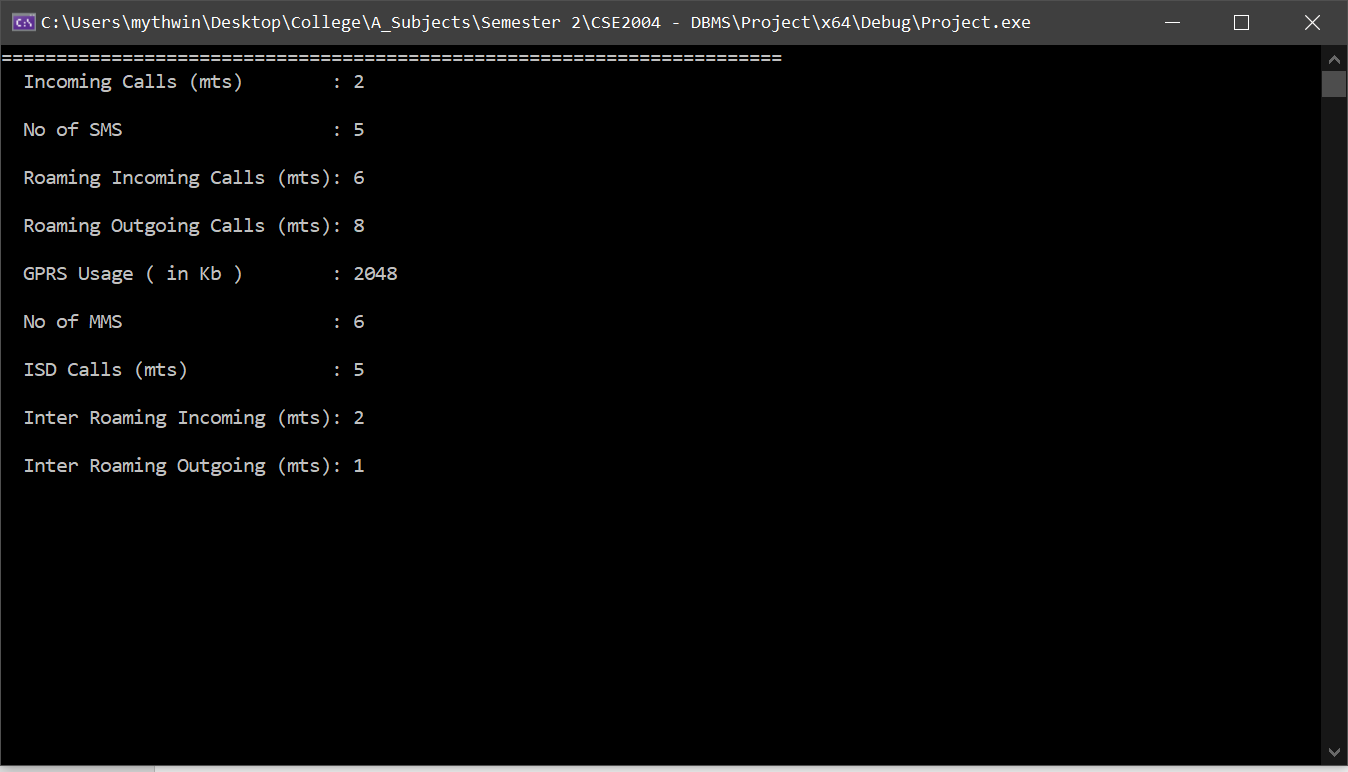


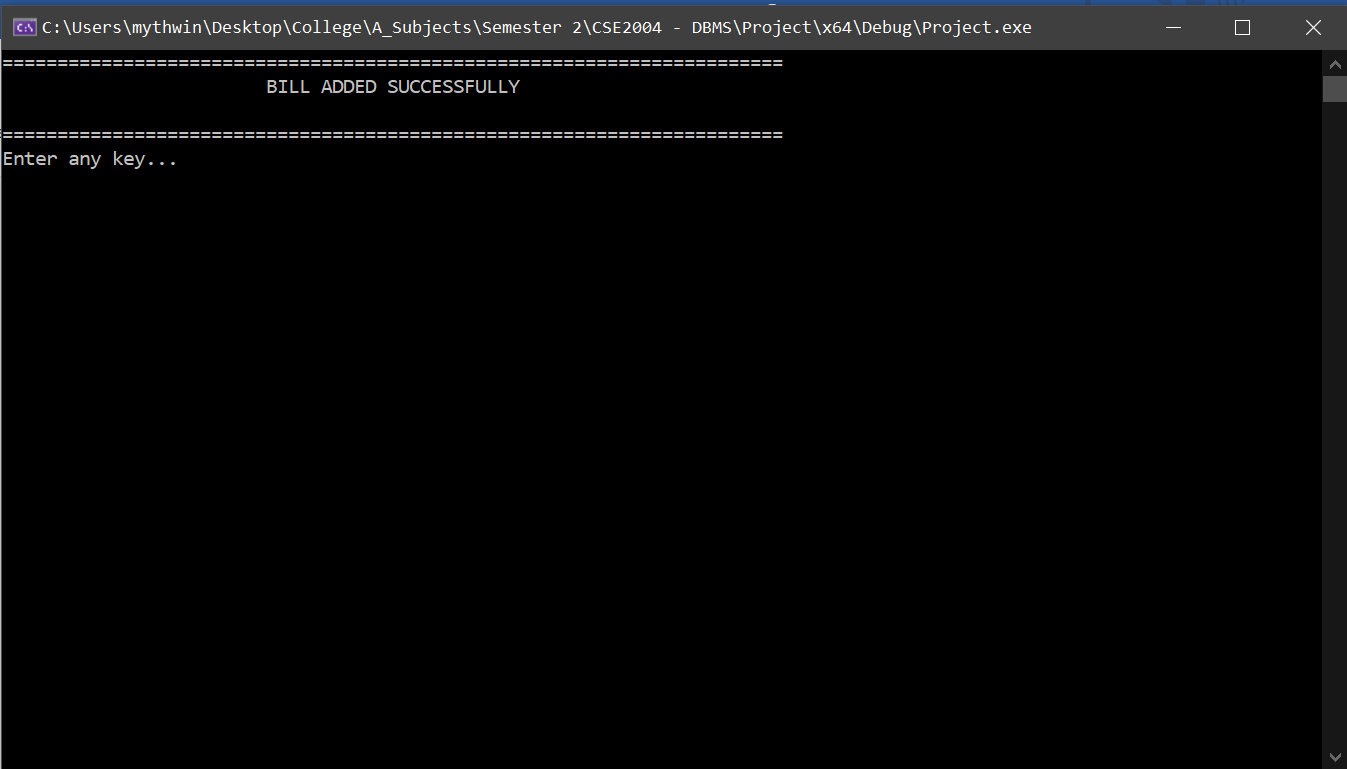


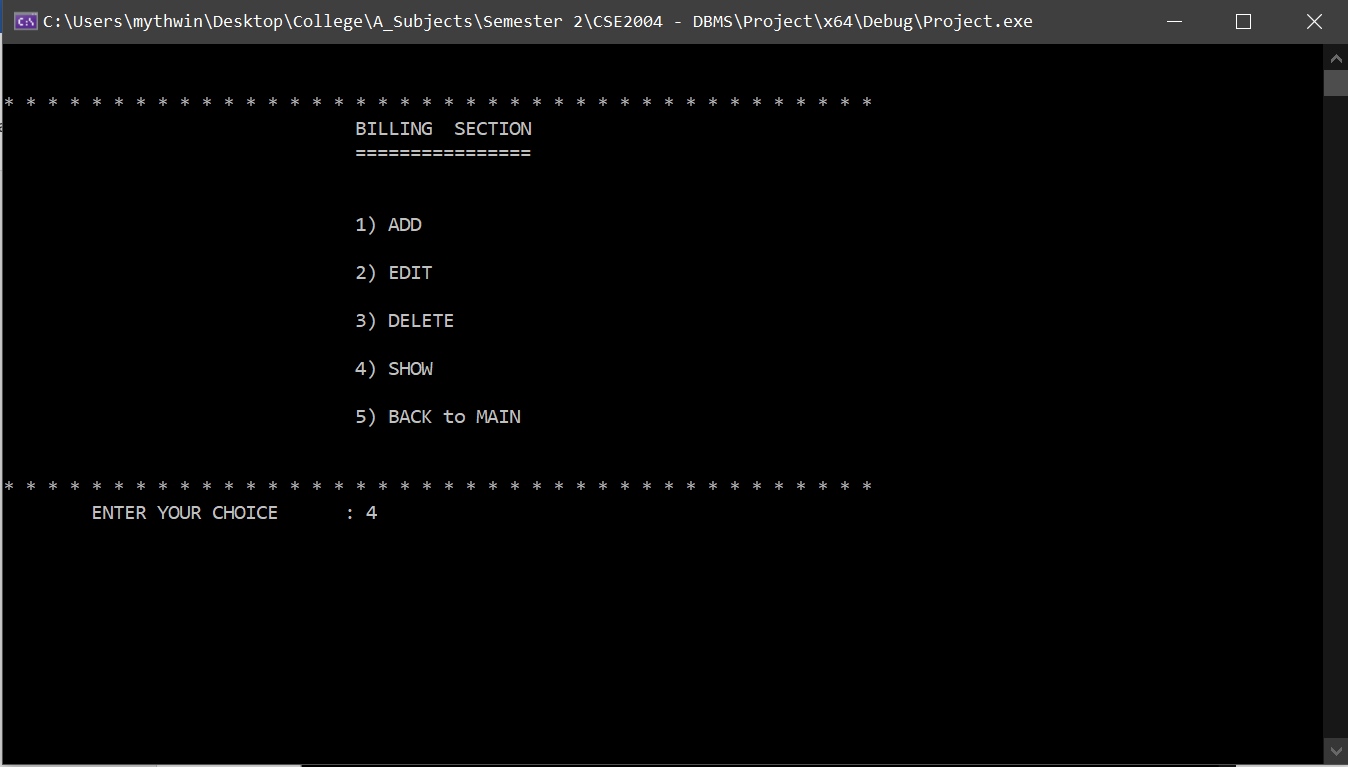


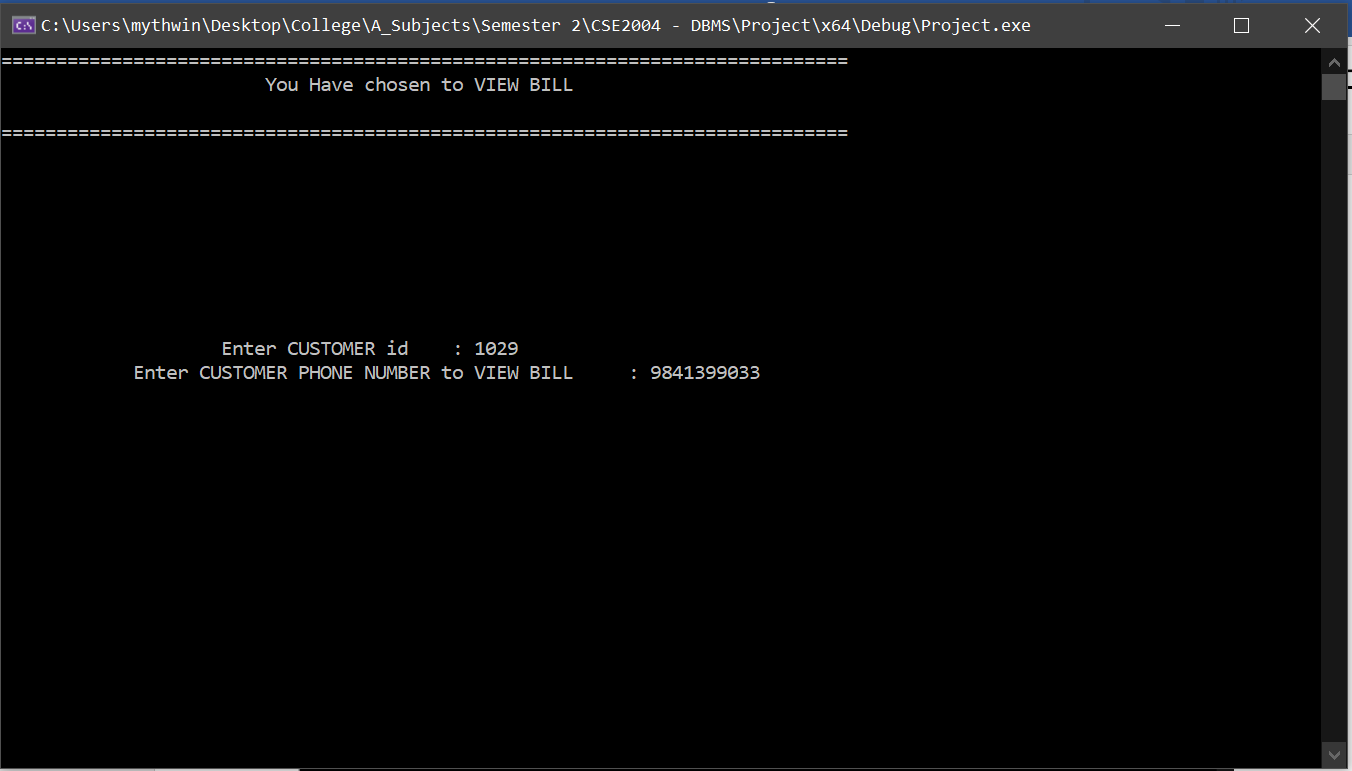


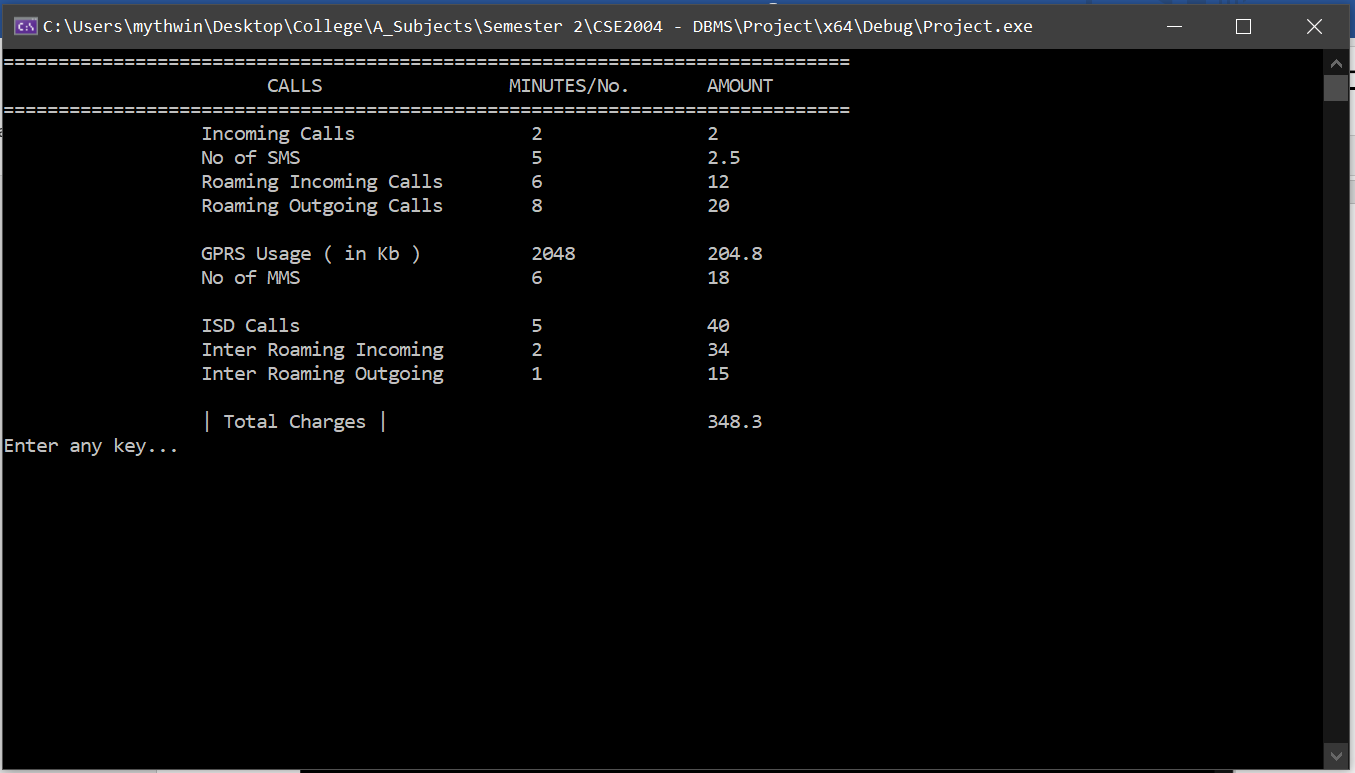


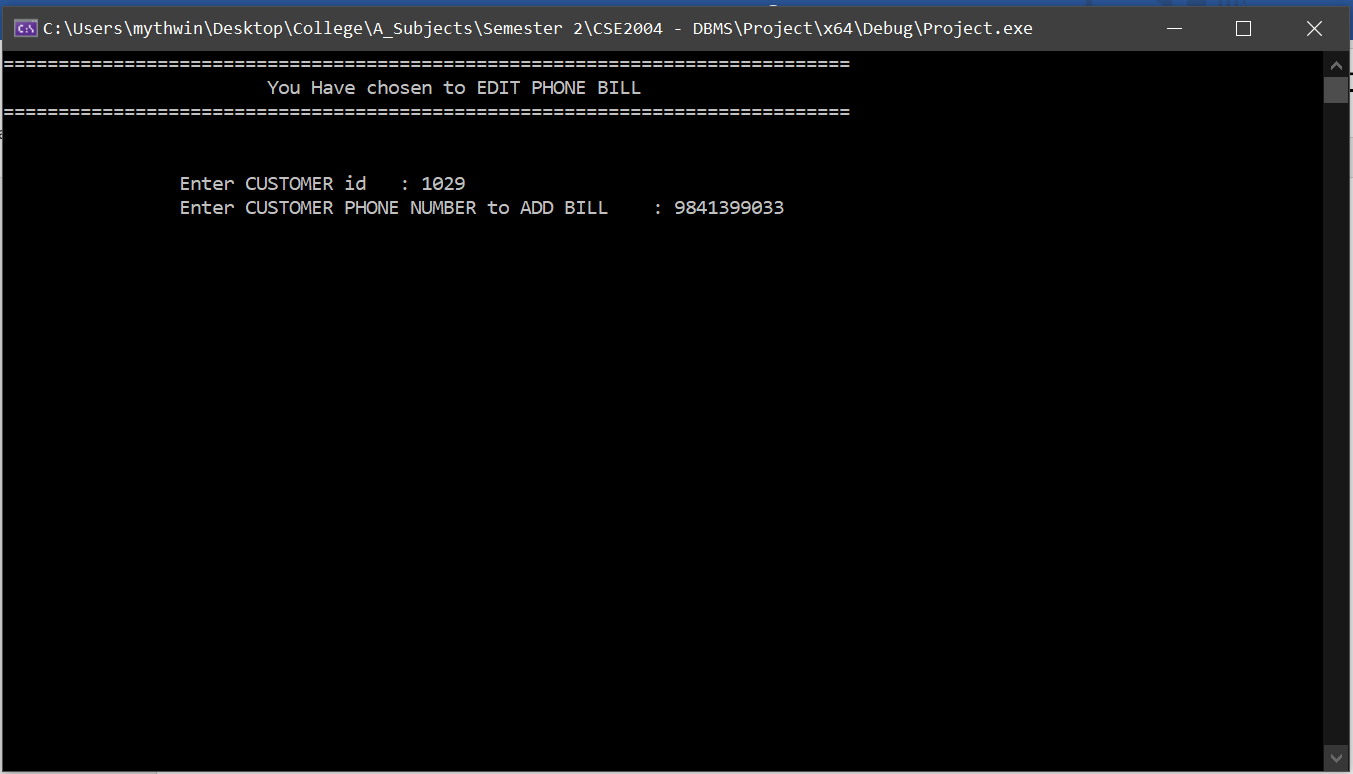


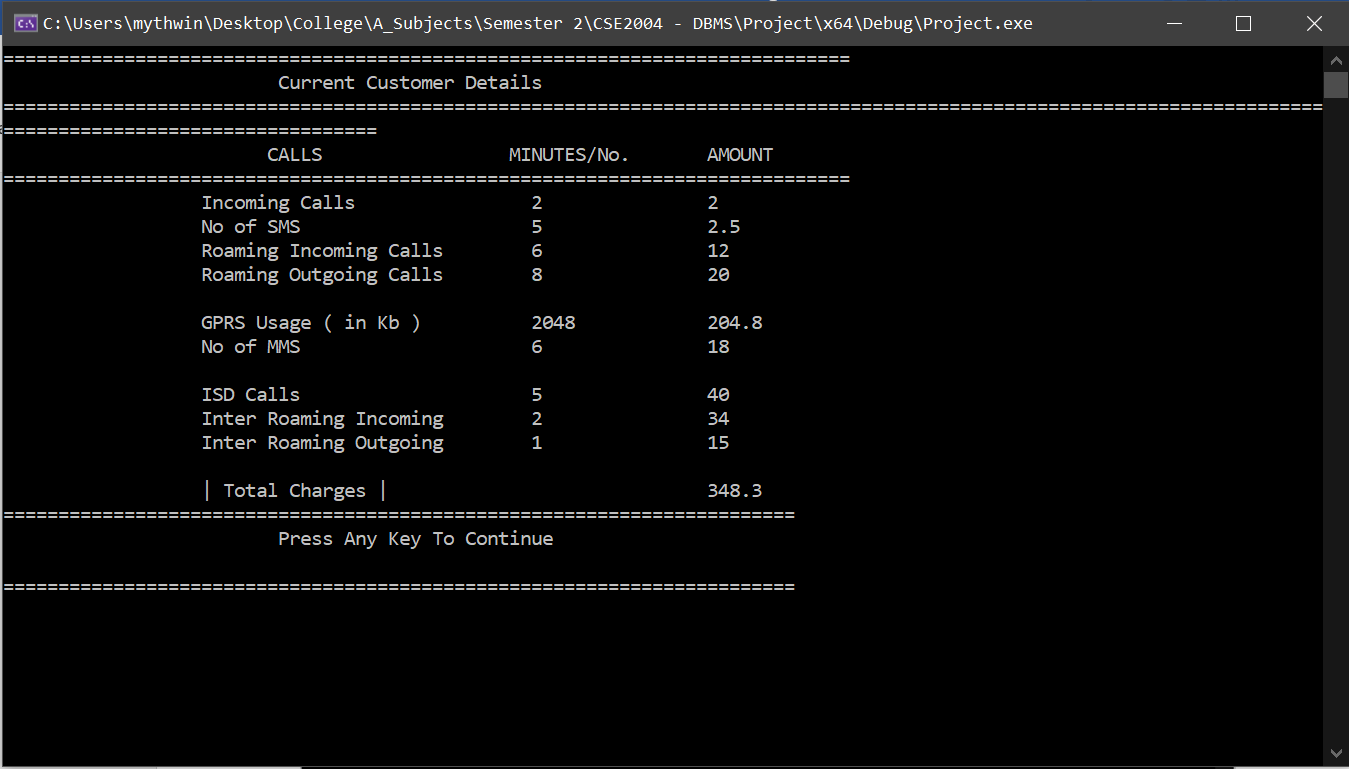


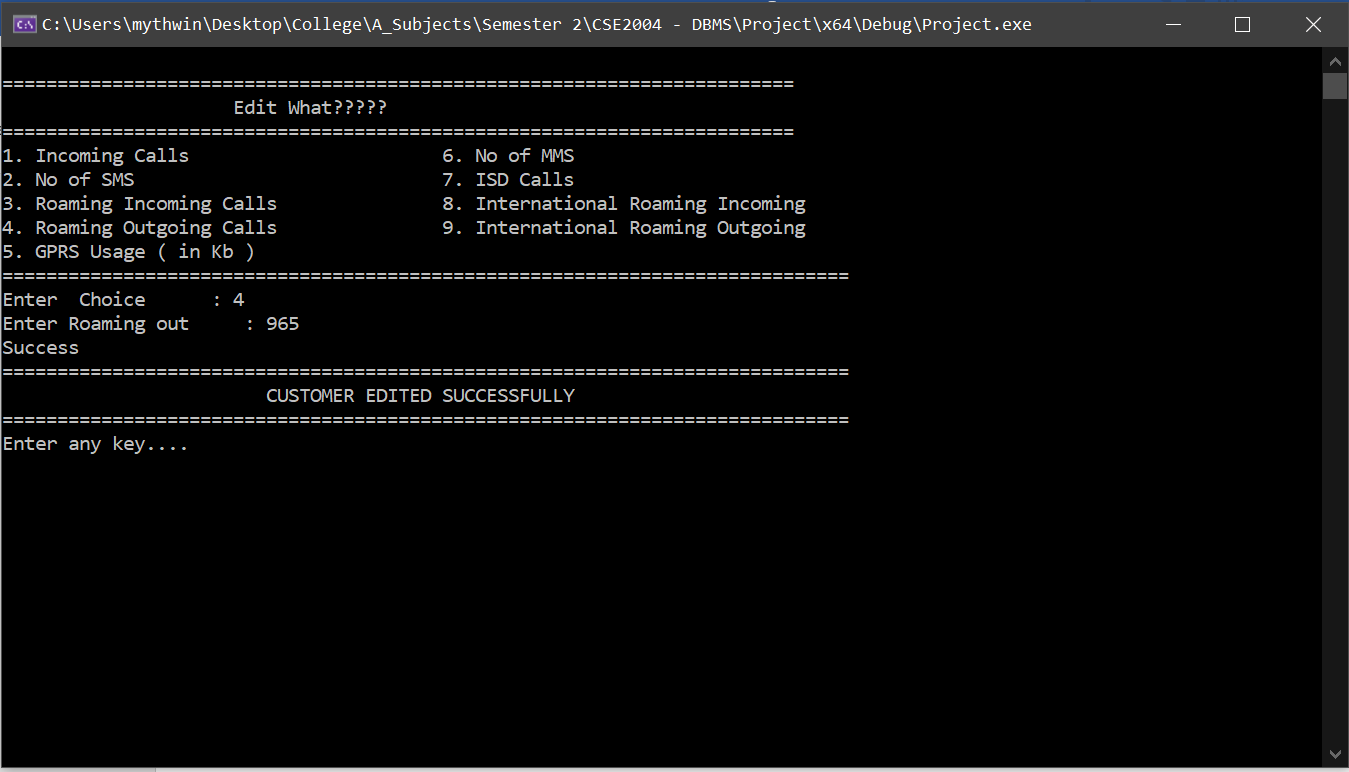


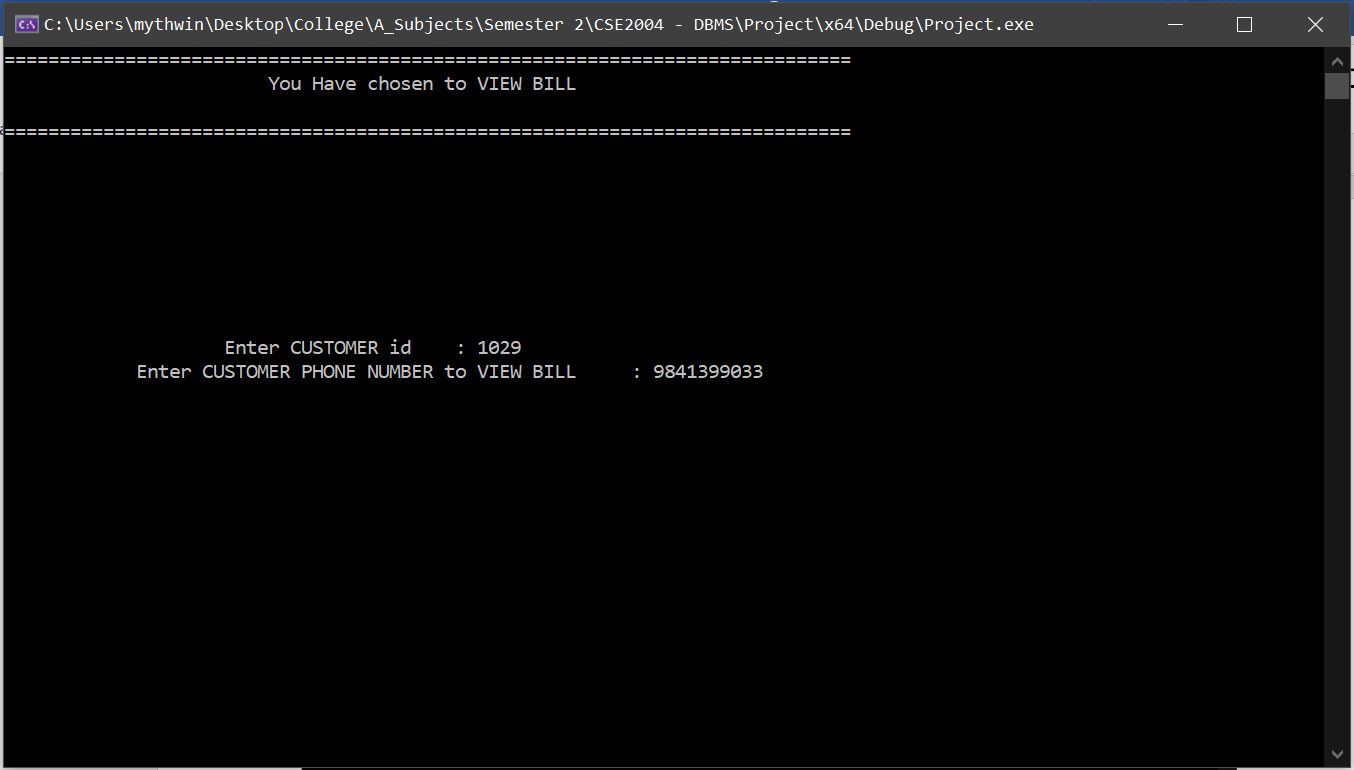


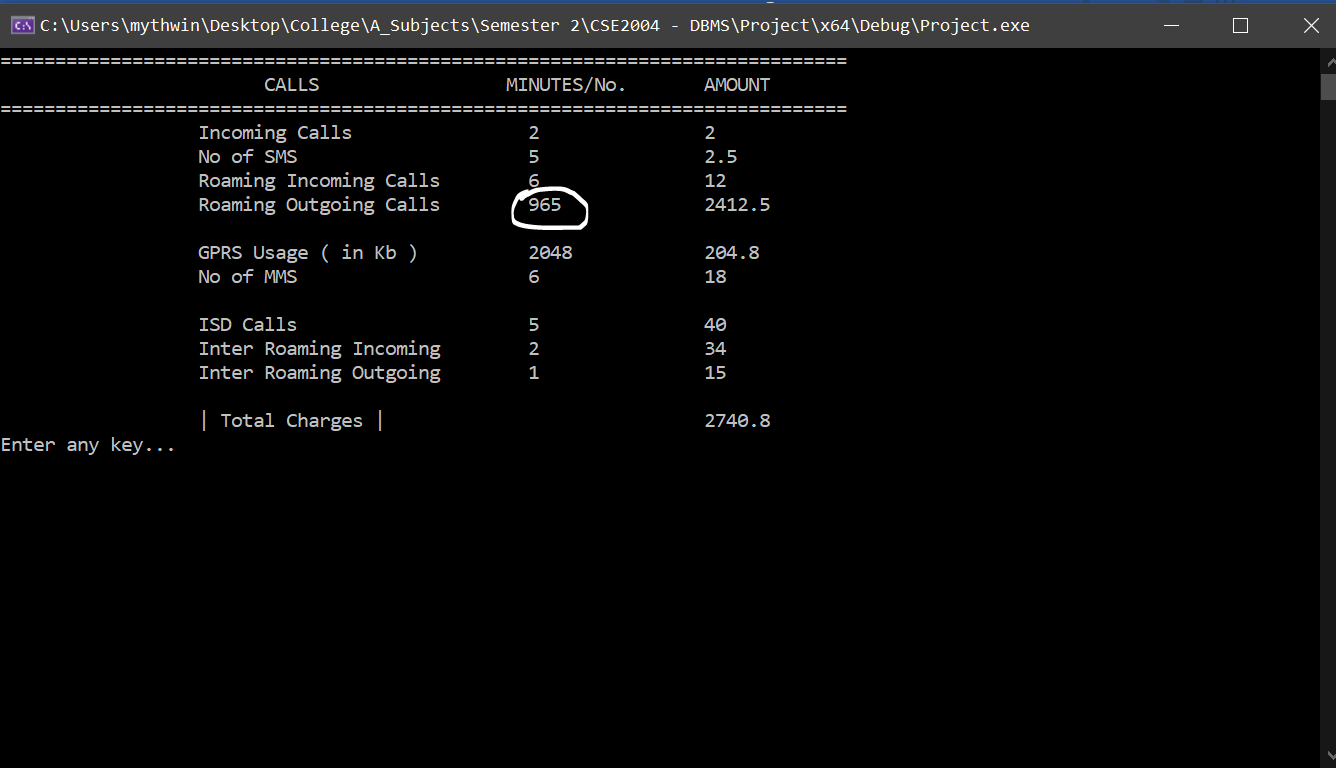


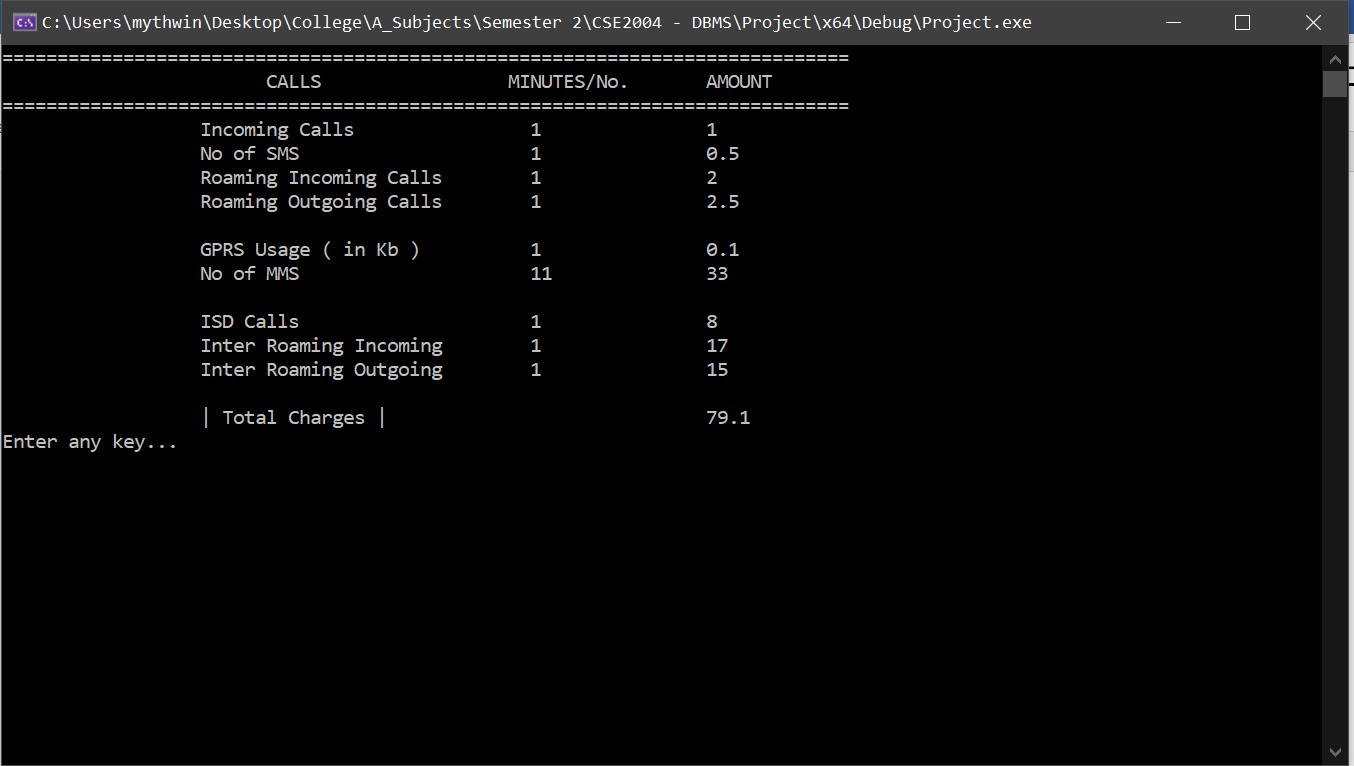


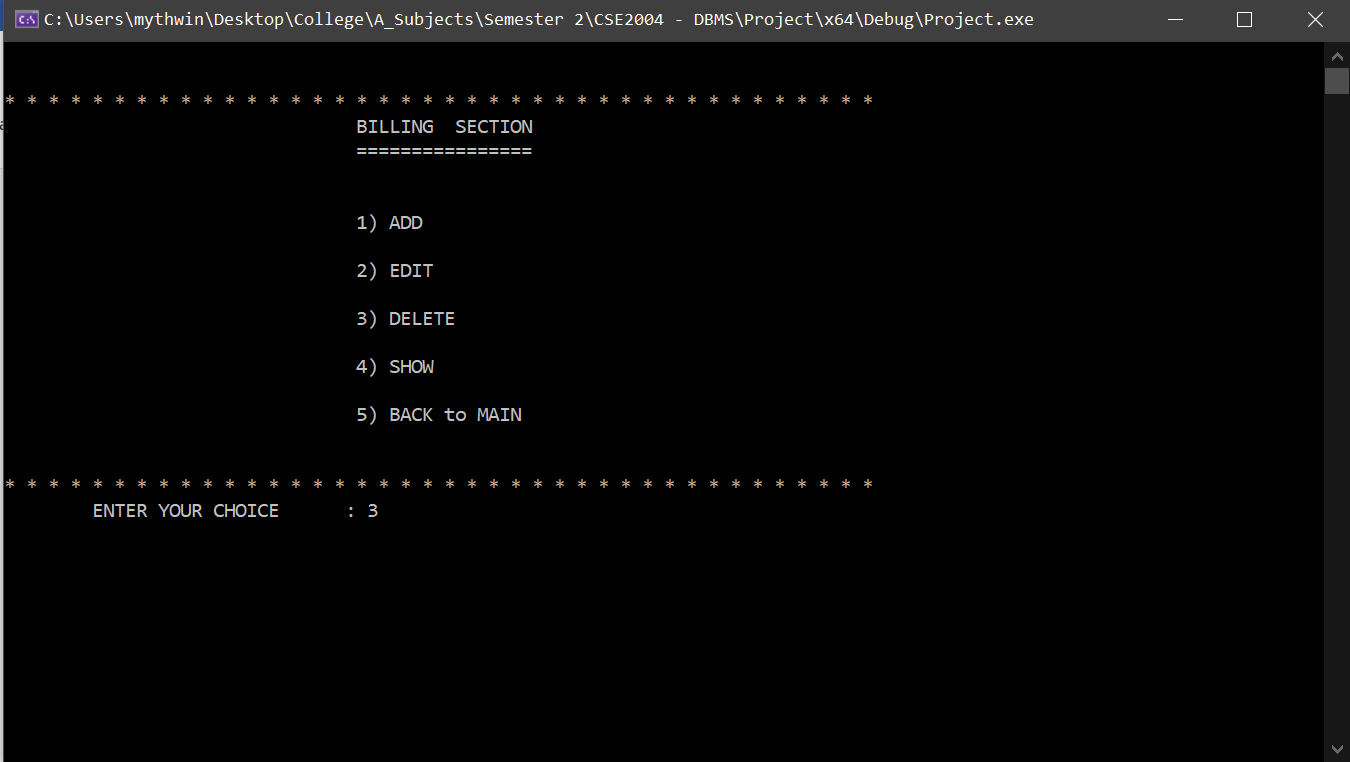


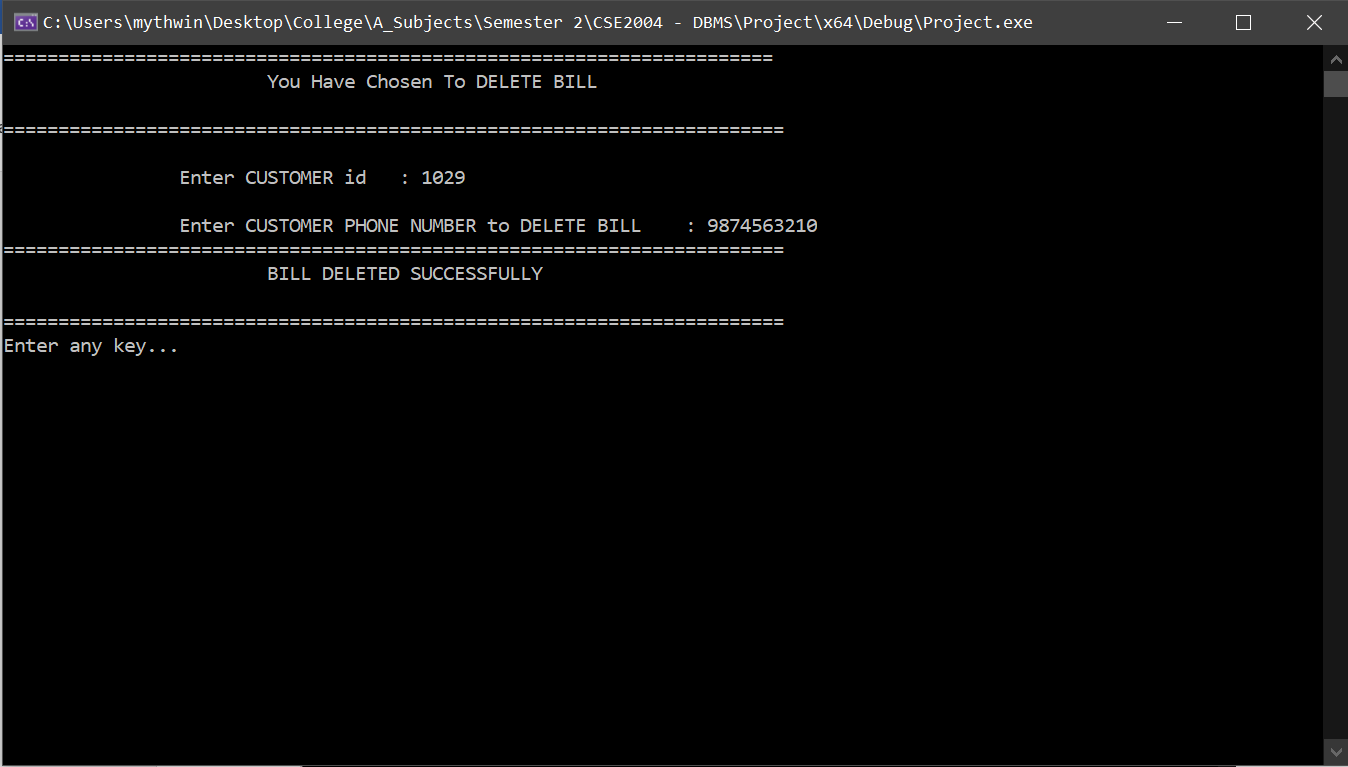


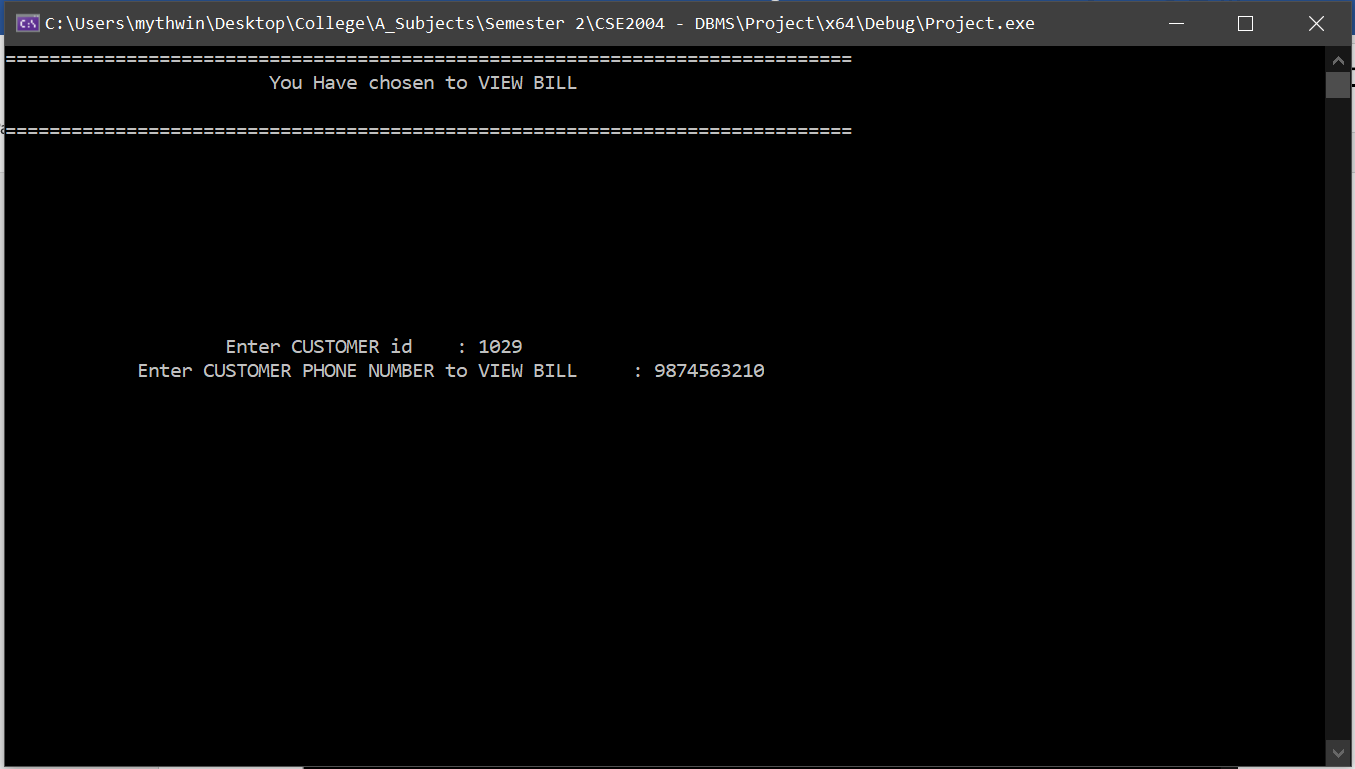


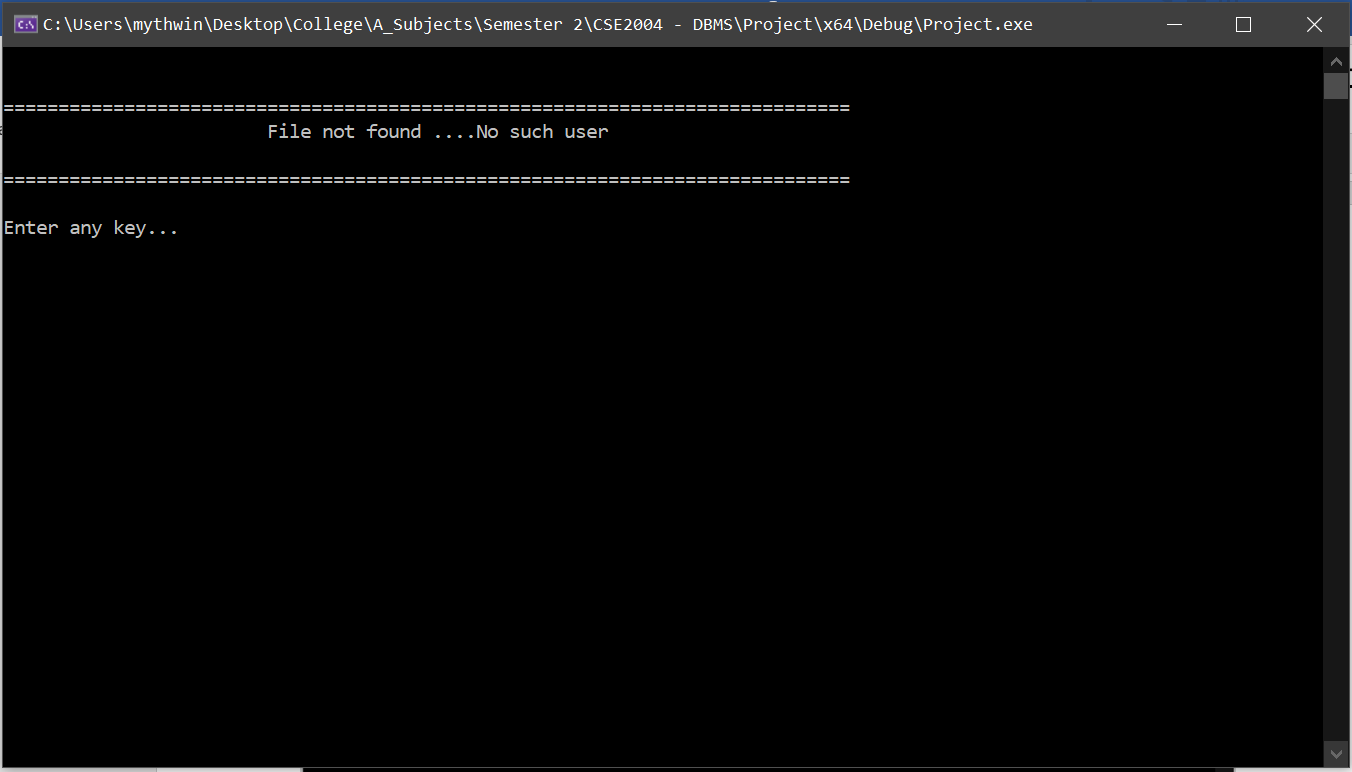


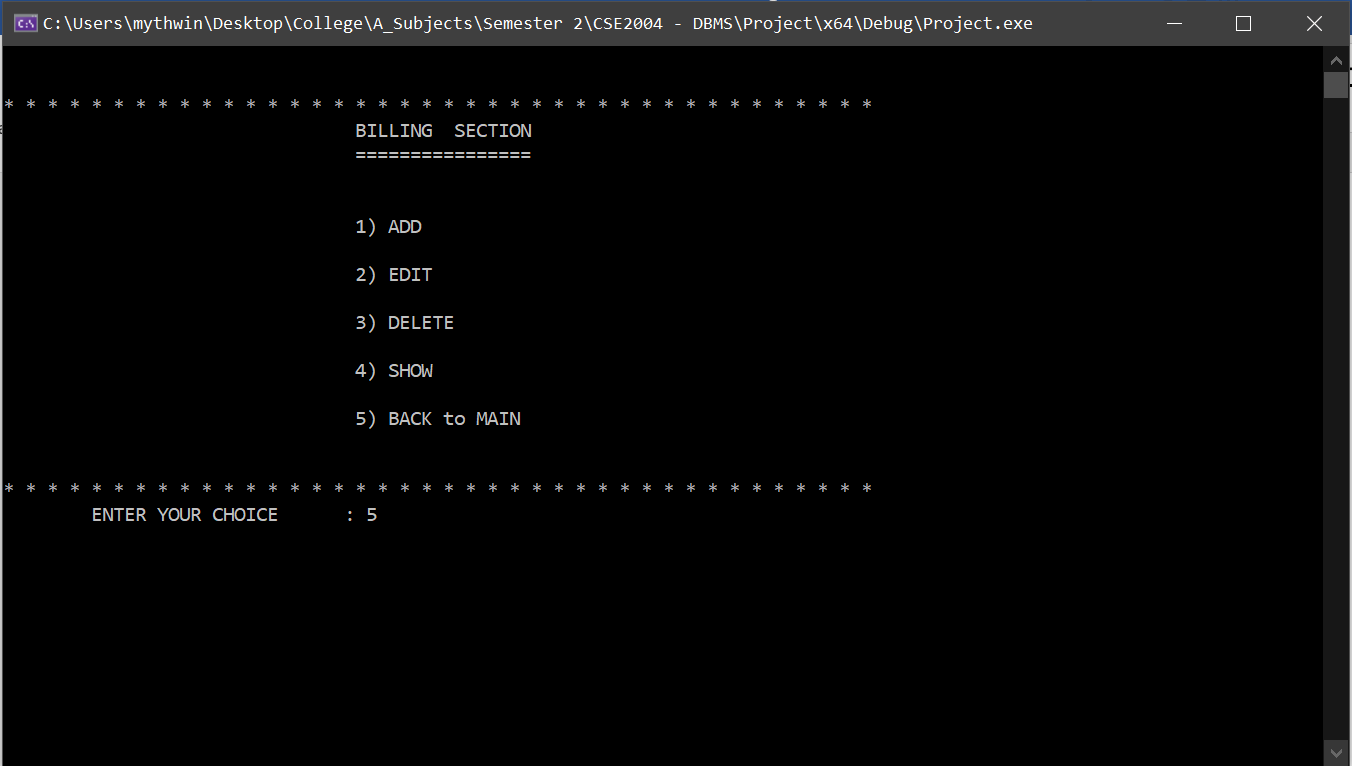


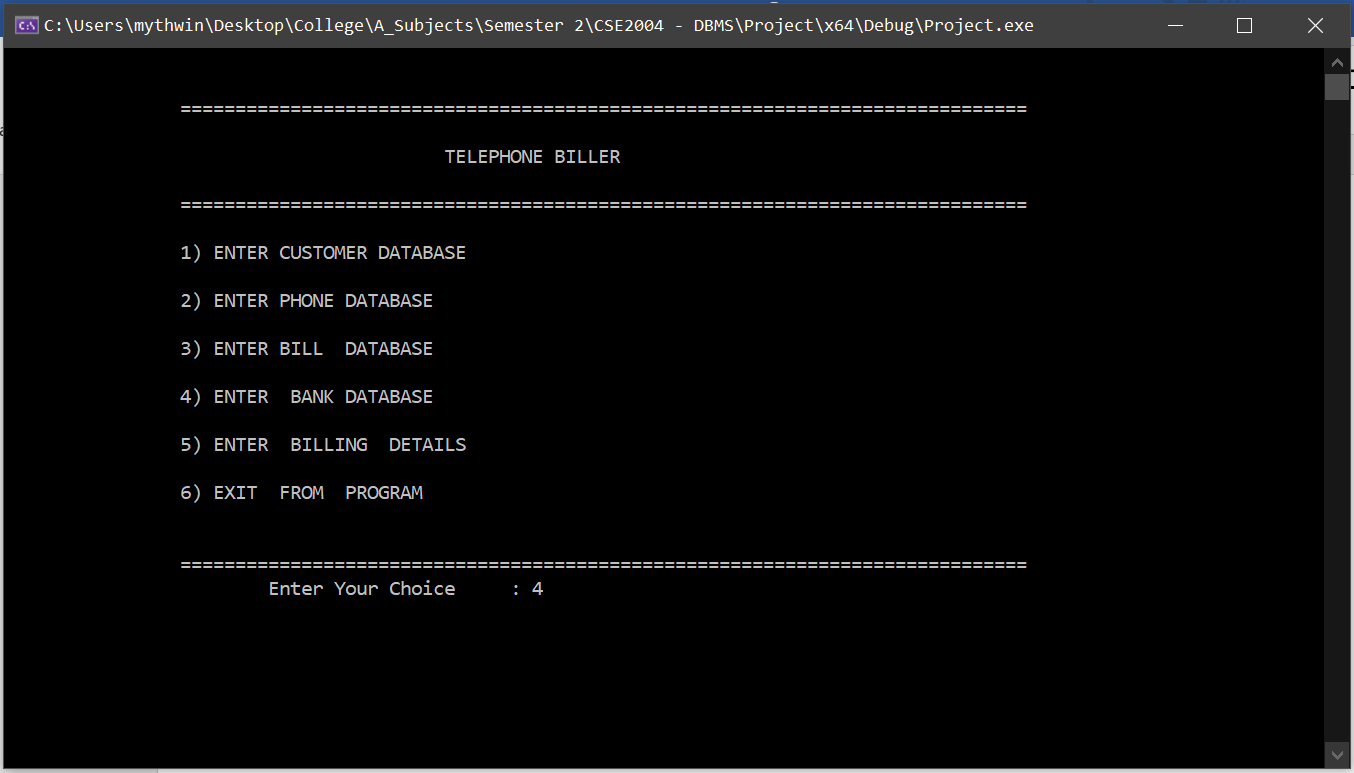


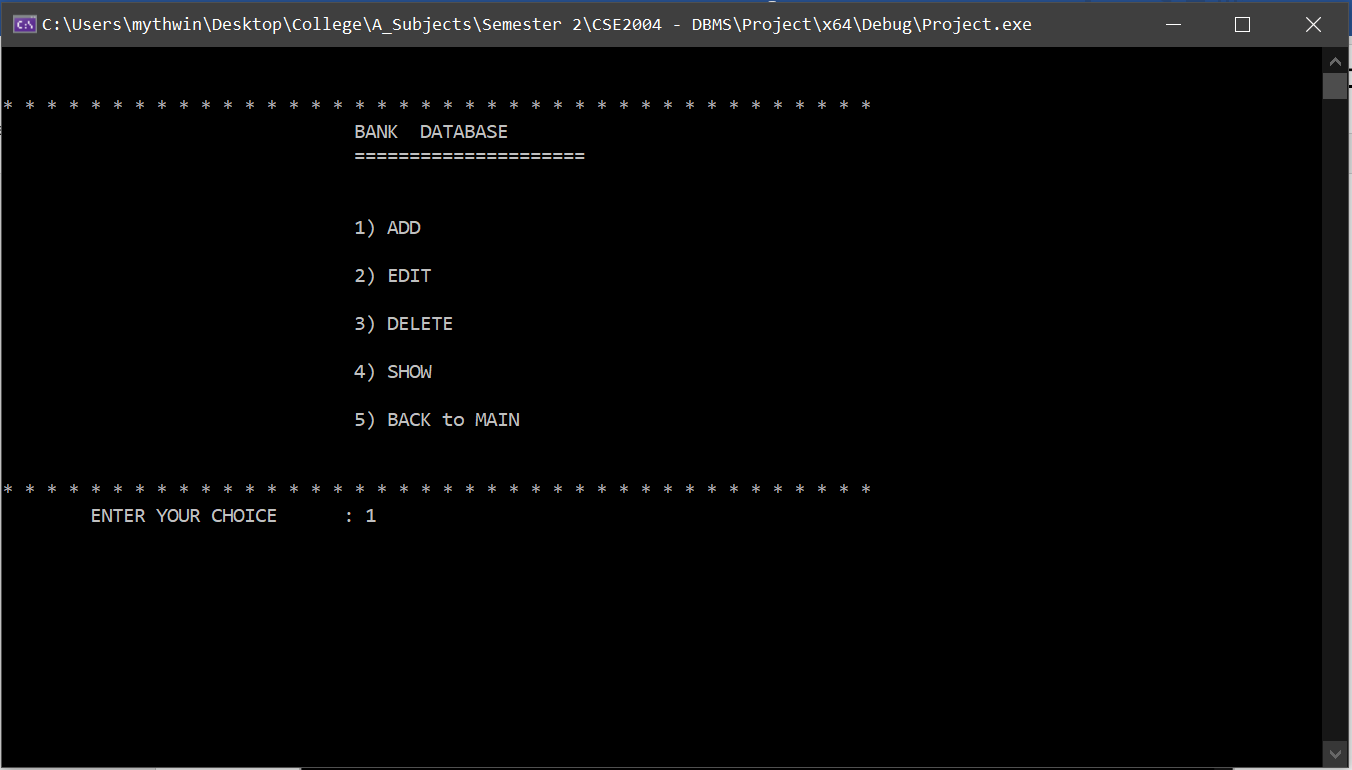


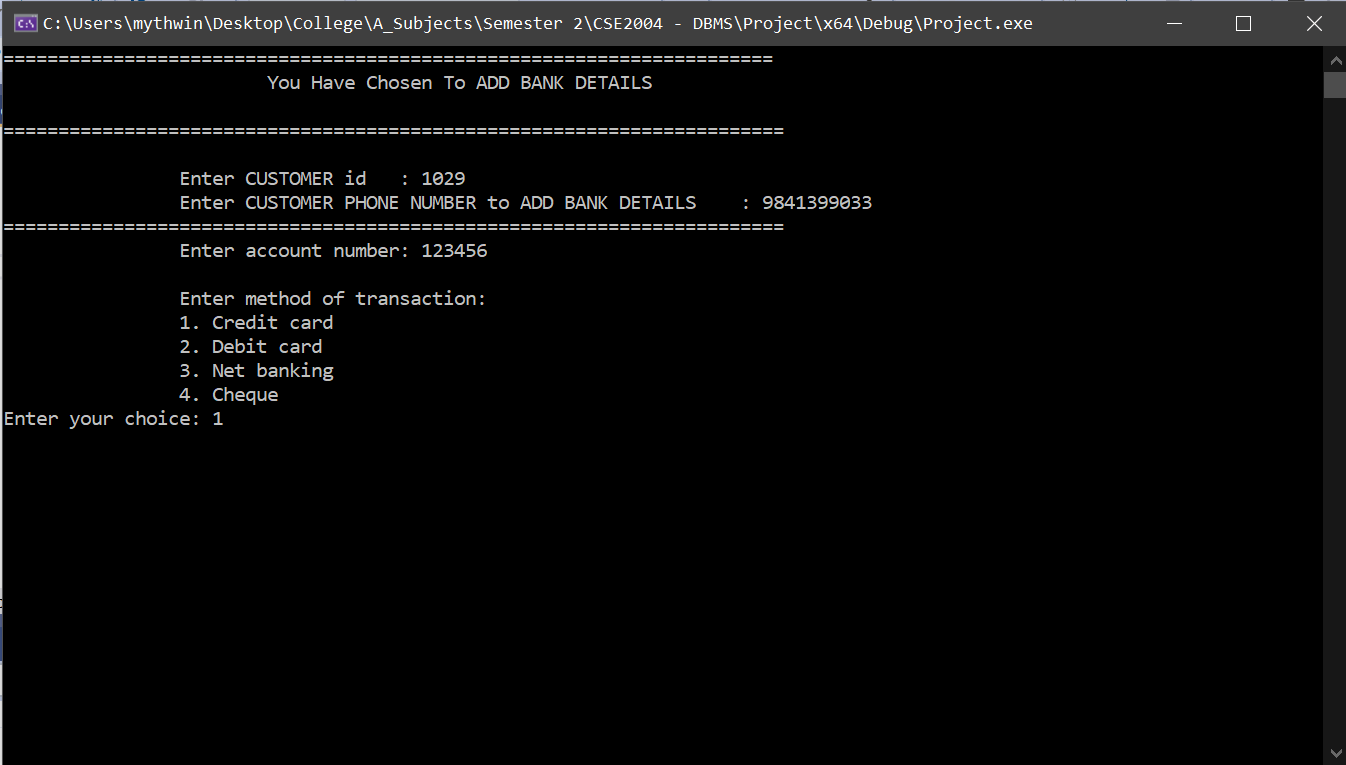


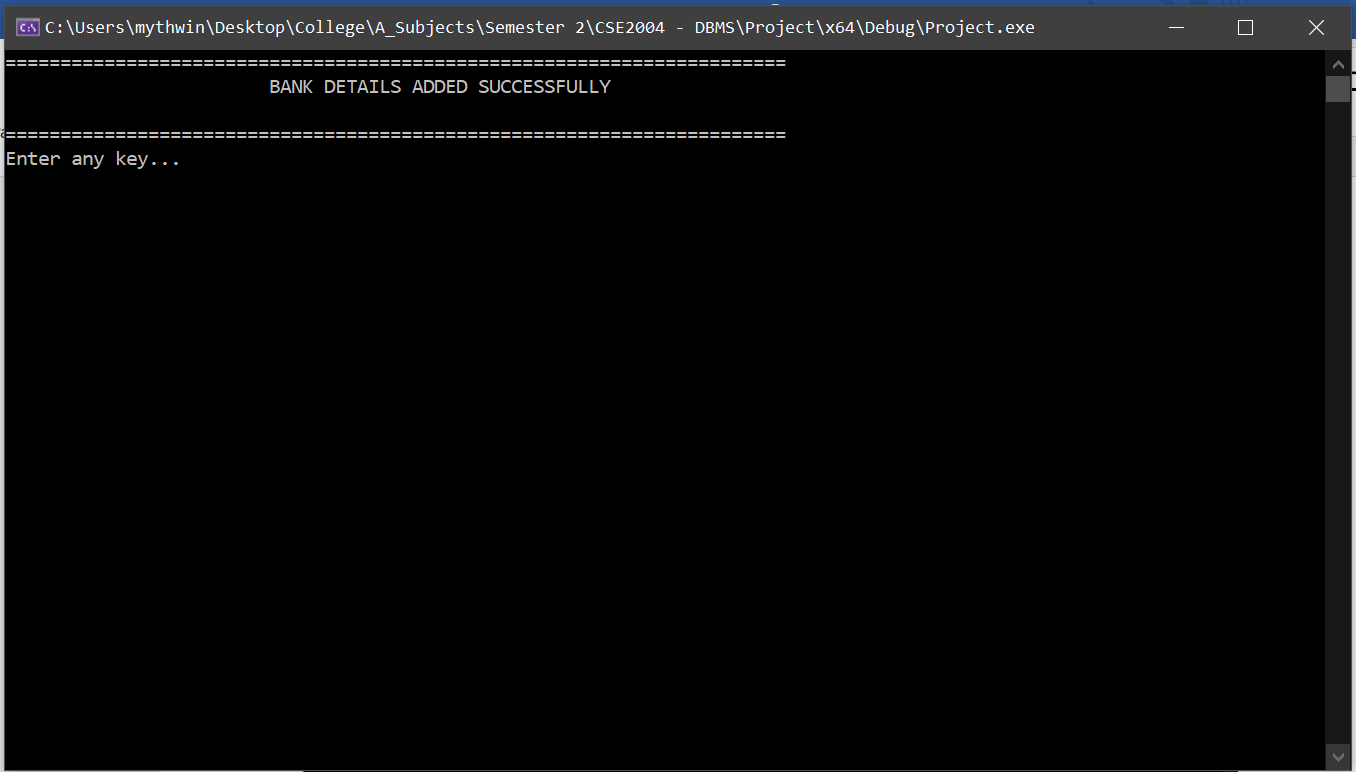


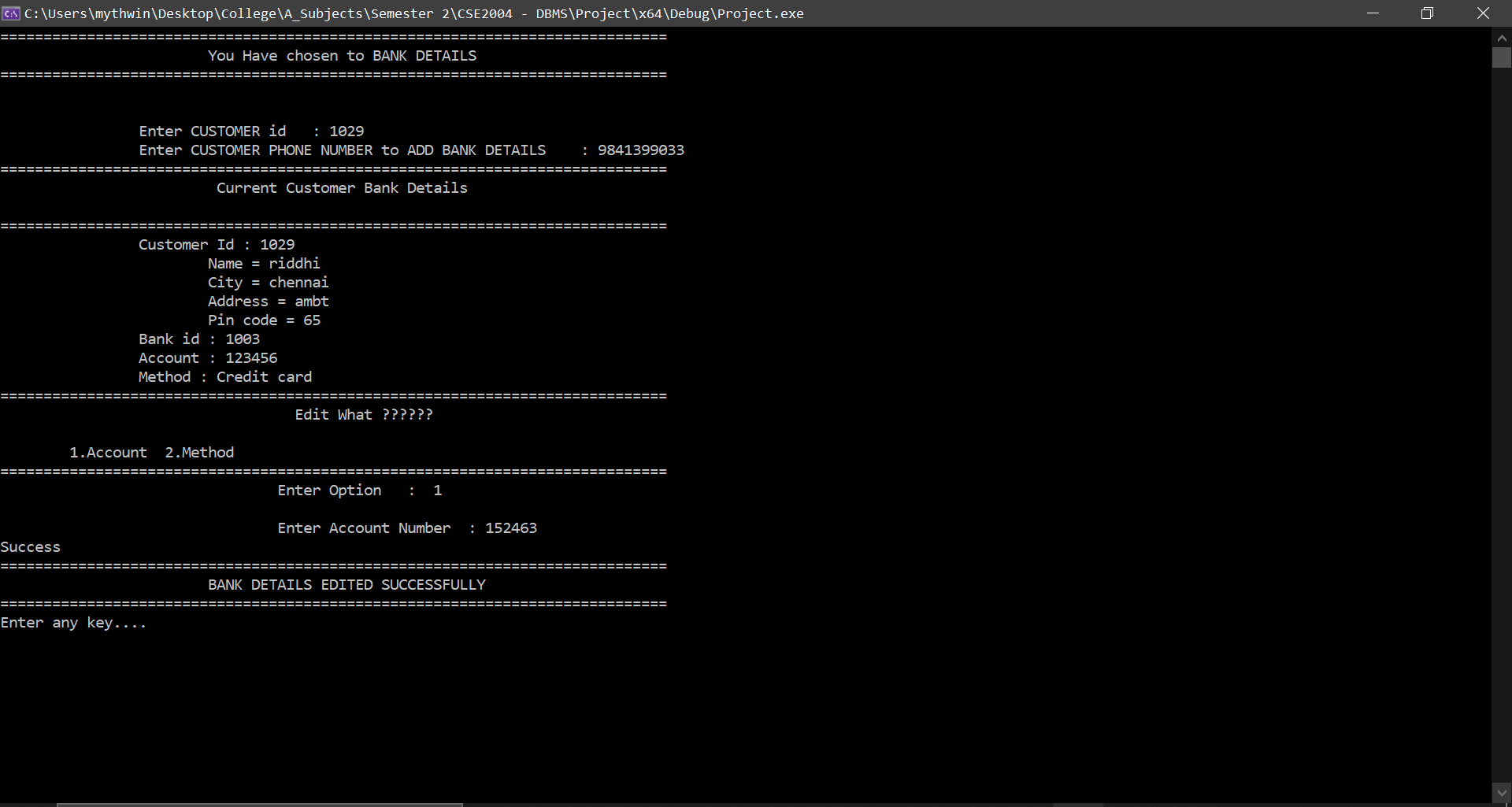


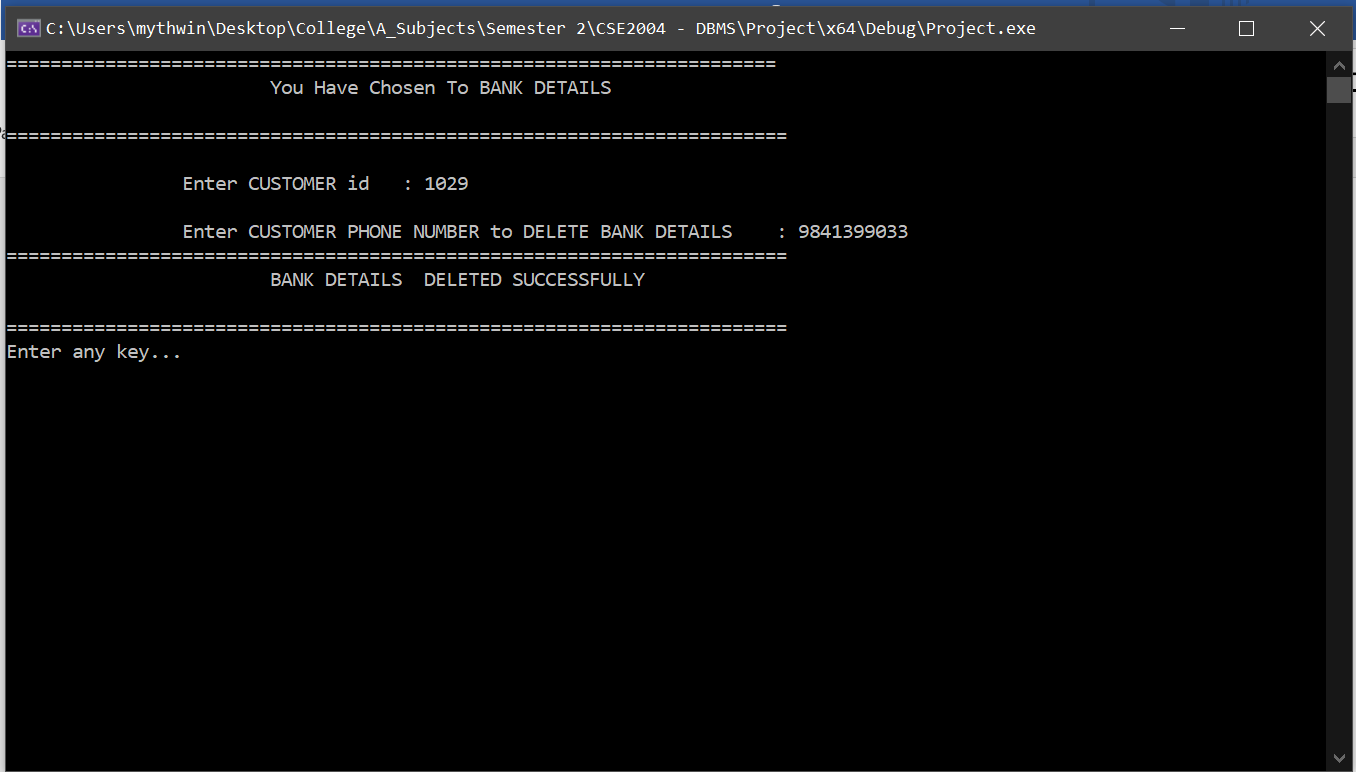


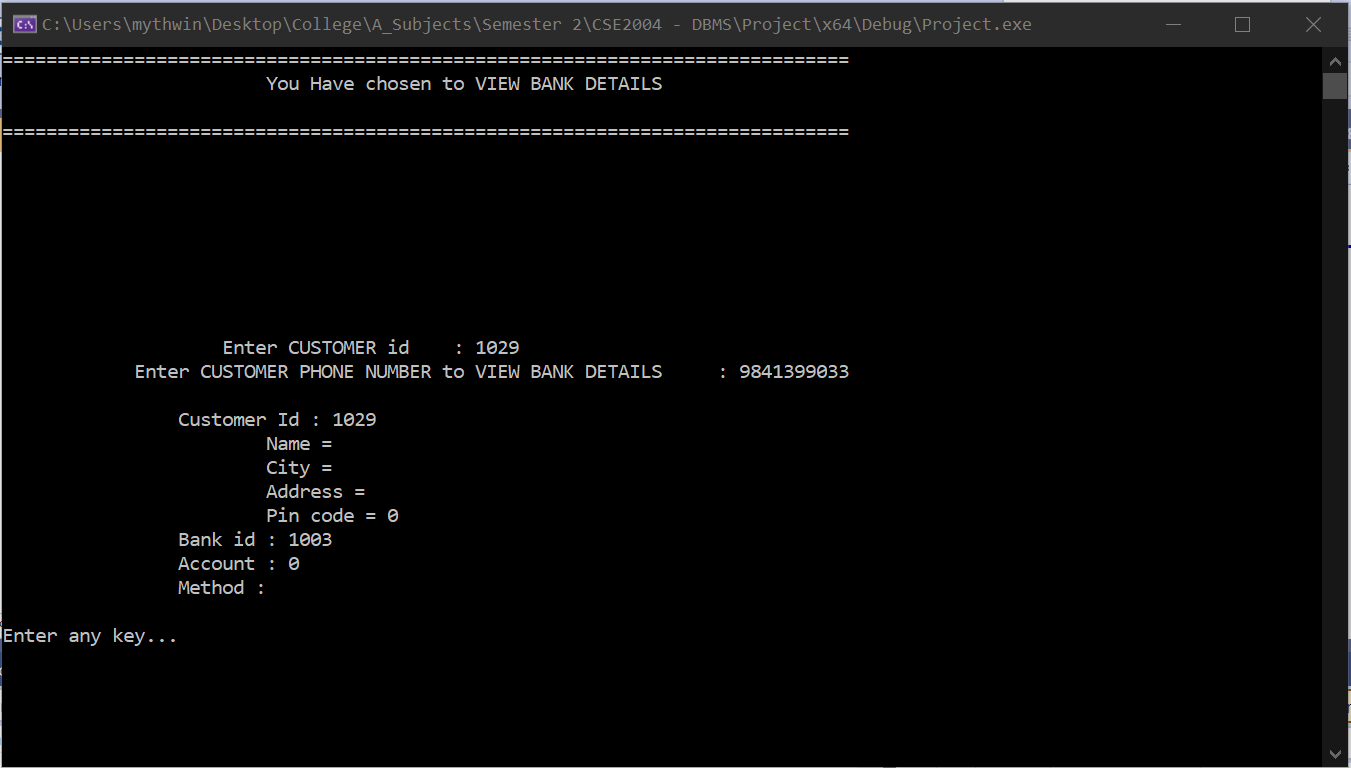


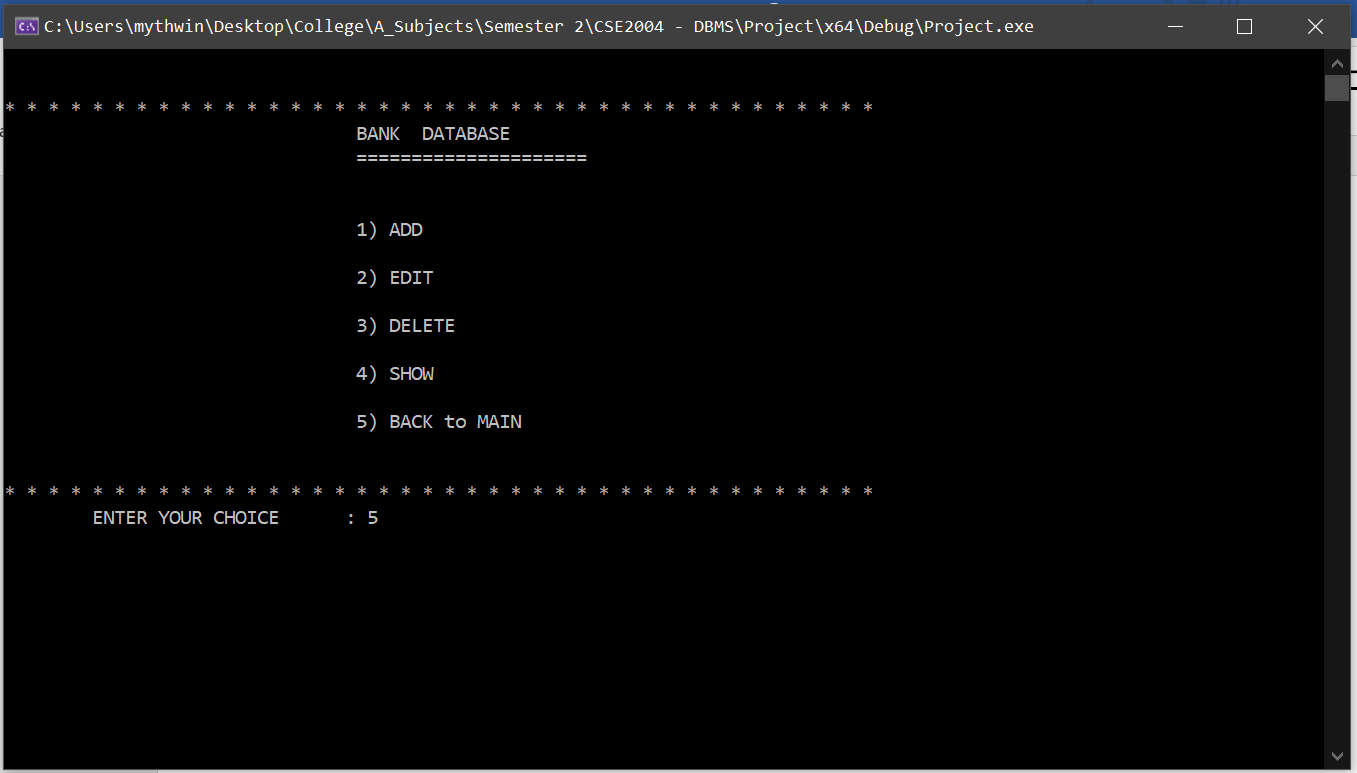


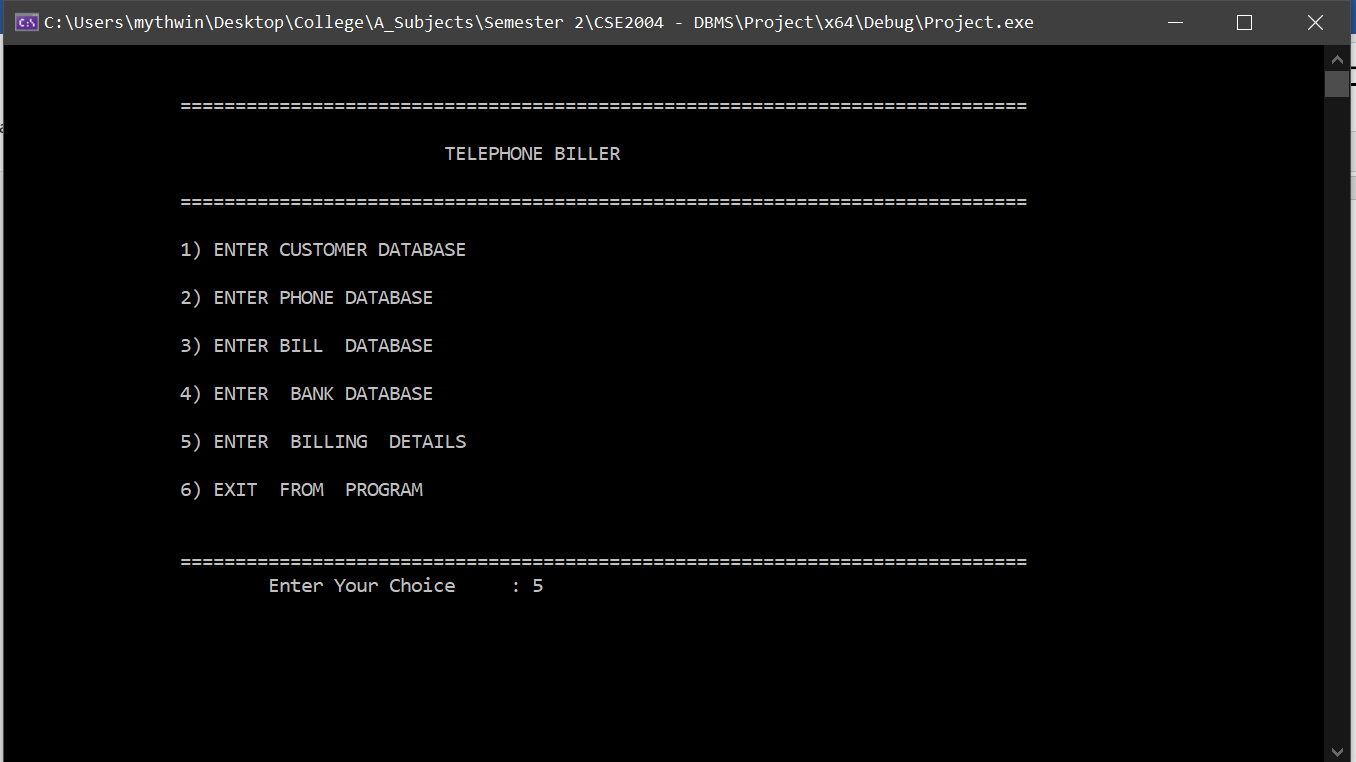


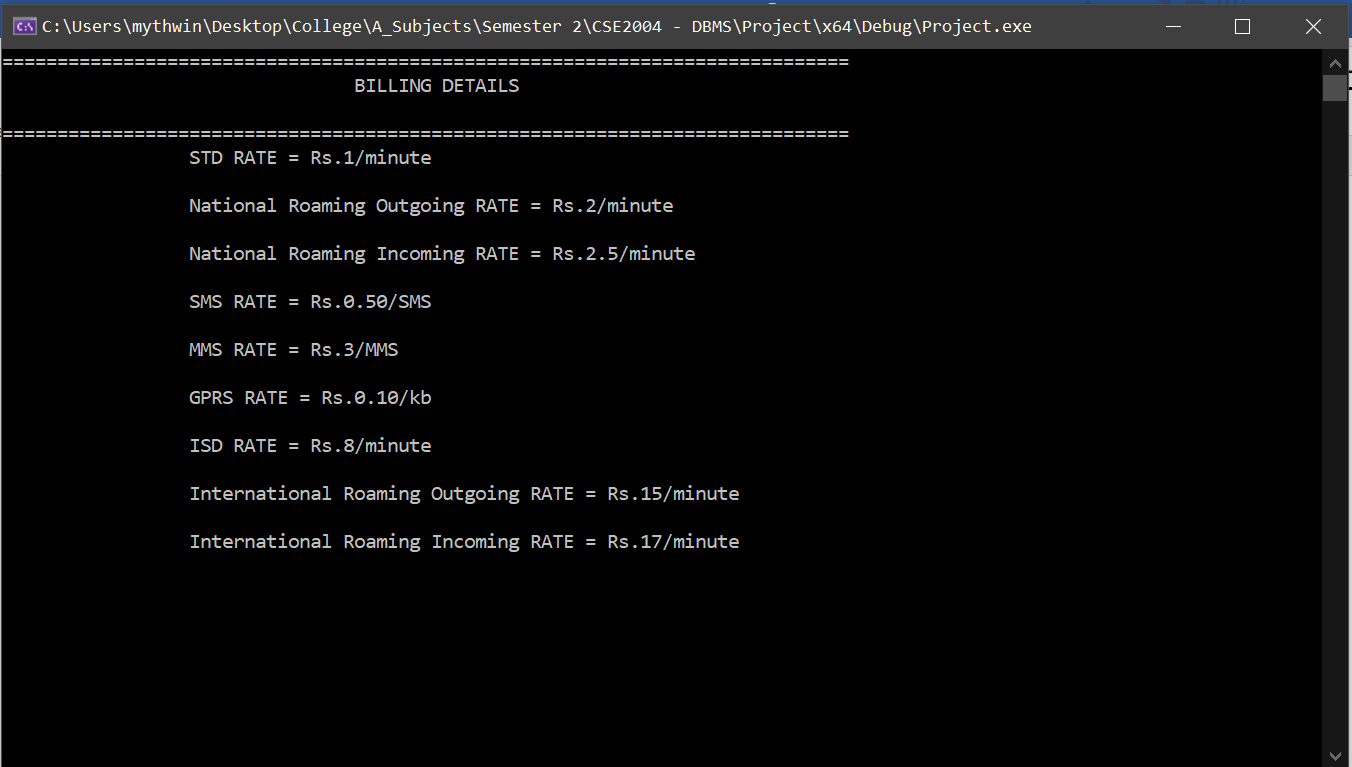


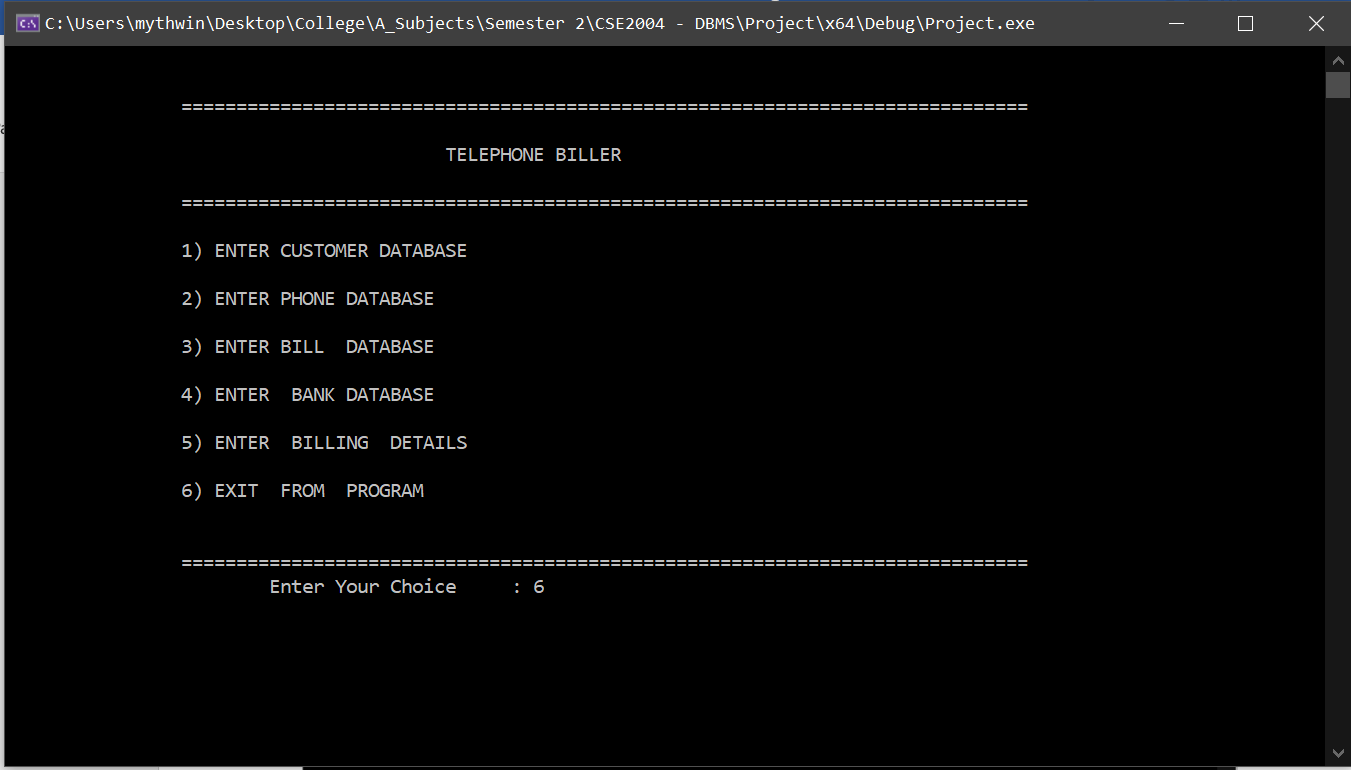


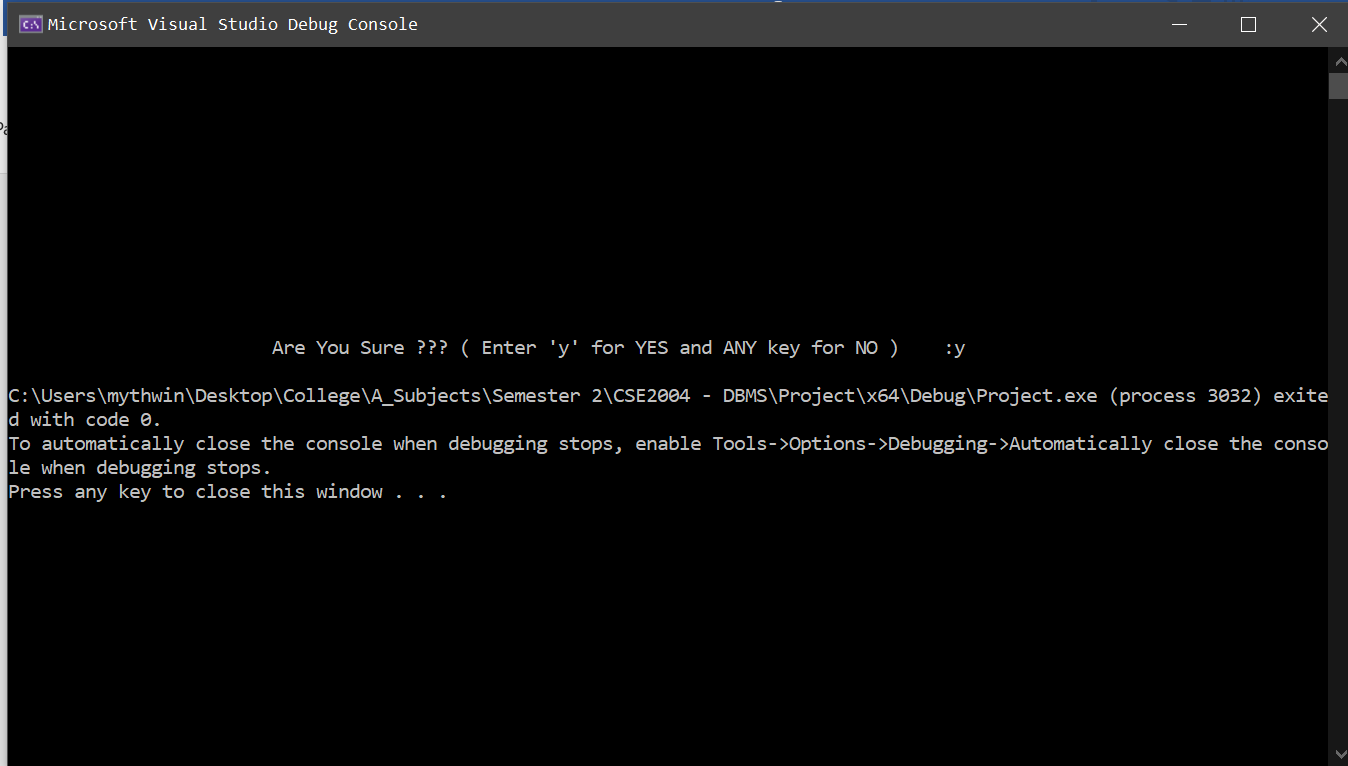












**Conclusion:**

In our project we have implemented our concepts so that it accurately reflects in our databases too. For example, if u delete customer details such as the name, address it will get deleted in our databases too without much delay or error.

It’s been made sure that all the datas that user enters gets stored in our databases and also, we have implemented some sequential codes to make sure that every data the user enter is unique and not duplicates. We have perfectly linked all our databases with the necessary primary keys and foreign keys.

Our code has not been boasted with colorful outputs and user interface yet a simpler, fantastic and efficient code.