```
 1: //BANKING MANAGEMENT SYSTEM WITH FILE HANDLING
 2: //Programmed by: BSCpE 2-1 Group 1
 3: //Esteban, Charlene Mae DG.
 4: //Gomez, Enrico Camilo P. Jr.
 5: //Guevarra, Justin Earl L.
 6: //Morita, Ami V.
 7: //Submitted to: Engr. Julius S. Cansino
 8:
 9: #include <iostream>
10: #include <fstream>
11: #include <cctype>
12: #include <iomanip>
13: #include <Windows.h>
14: #include <string>
15: #include <time.h>
16: #include <cstdlib>
17: #include <conio.h>
18: using namespace std;
19:
20: int usernum;
21: struct account
22: {
23:
24:     int accNo;
25:     int accPin;
26:     char accName[50];
27:     float accBalance;
28:     char accType;
29:     void createAccount();
30:     void show() const;
31:     void modify(int);
32:     void deposit(int);
33:     void withdraw(int);
34:     void accDetails(int) const;
35:     int retaccPin() const;
36:     int retaccNo() const;
37:     int retaccBalance() const;
38:     char retaccType() const;
39: }ac;
40:
41: //Functions needed in file handling******************
42: void writeAccount();    //function to write record in binary file
43: void displayAccount(int);   //function to display account details given by user
44: void modifyAccount(int,int);    //function to modify record of file
45: void deleteAccount(int);    //function to delete record of file
46: void dispAll();     //function to display all account details
47: void depwith(int, int); // function to desposit/withdraw amount for given account
48: void welcome();
49: void logoandtime();
50: int login();
51: void matchAccount(int);
52:
53: //FUNCTIONS
54: void pos(int x, int y){
55:     COORD coord;
```

```
 56:     coord.X = x;
 57:     coord.Y = y;
 58:     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord);
 59: }
 60:
 61: void col(int k){
 62:     HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
 63:     SetConsoleTextAttribute(hConsole, k);
 64: }
 65:
 66: const std::string currentDateTime() {
 67:     time_t     now = time(0);
 68:     struct tm  tstruct;
 69:     char       buf[80];
 70:     tstruct = *localtime(&now);
 71:     strftime(buf, sizeof(buf), "%X   %B %d, %Y", &tstruct);
 72:     return buf;
 73: }
 74:
 75: void logoandtime(){
 76:     col(7);pos(1,1);cout <<currentDateTime();
 77:     col(205);pos(90,1);cout<<"    ";
 78:     pos(90,2);cout<<" ";
 79:     pos(90,3);cout<<"    ";
 80:     pos(92,2);cout<<" ";
 81:     pos(94,1);cout<<" ";
 82:     pos(94,2);cout<<" ";
 83:     pos(94,3);cout<<"    ";
 84:     col(240);pos(90,4); cout<< "B A N K";
 85:     col(15);
 86: }
 87:
 88: void account::createAccount()
 89: {
 90:     logoandtime();
 91:     col(112);cout<<"\n\tAccount No.: ";
 92:     accNo=rand()%1000 + 1500; //for random making of account numbers
 93:
 94:     bool found=false;
 95:
 96:     fstream File;
 97:     File.open("accdata.dat",ios::binary|ios::in|ios::out);
 98:     while(!File.eof() && found==false)
 99:     {
100:         File.read(reinterpret_cast<char *> (&ac), sizeof(account));
101:         if(ac.retaccNo()==accNo)
102:         {
103:                 accNo=rand()%1000 + 1500;
104:           }
105:     }
106:     File.close();
107:
108:     cout<<accNo;
109:         col(15);
110:
```

```
111:     int tempPin1, tempPin2, loop1;
112:     cout << "\n\tEnter your PIN: ";
113:     cin >> tempPin1;
114:     do{
115:         loop1=1;
116:         cout <<"\tRetype PIN: ";
117:         cin >>tempPin2;
118:         if(tempPin1==tempPin2)
119:             accPin=tempPin1;
120:         else loop1=0;
121:     }while(loop1==0);
122:
123:
124:     int x=0,loop=0;
125:     char tempaccName[50];
126:     do
127:     {
128:         cout<<"\n\n\tFull Name: ";
129:         cin.ignore();
130:         cin.getline(tempaccName,50);
131:         x=0;
132:         for(int i=0 ; i < strlen(tempaccName); i++){
133:             if(!isalpha(tempaccName[i]) ){
134:                 loop = 1;
135:                 x++;
136:                 if(isspace(tempaccName[i])){
137:                     x--;
138:                 }
139:                 if(ispunct(tempaccName[i])){
140:                     x--;
141:                 }
142:             }
143:         }
144:         if(x==0){
145:             strcpy(accName, tempaccName);
146:             loop = 0;
147:         }
148:         else {
149:             col(12);cout <<"\tInvalid Name!";col(15);
150:         }
151:     }while(loop == 1);
152:
153:     x=1;
154:     do{
155:         cout<<"\n\tEnter Type of Account (C/S) Current or Savings: ";
156:         accType=_getch();
157:         accType=toupper(accType);
158:         cout <<accType<<endl;
159:         if((accType == 'S')||(accType=='C'))
160:             x=0;
161:         else {
162:             col(12);cout << "\tPlease Enter a Valid Type of Account!";col(15);}
163:     }while(x==1);
164:     loop=0;
165:     do{
```

```
166:     cout<<"\n\tEnter the Initial Deposit Amount: Php ";
167:     cin>>accBalance;
168:     if ((accType=='S')&&(accBalance<500))
169:     {
170:             col(12);cout<<"\n\tRequired Amount for Savings Account is Php500.00";col(15);
171:             loop=1;
172:     }
173:     else if((accType=='C')&&(accBalance<1000))
174:     {
175:         col(12);cout<<"\n\tRequired Amount for Current Account is Php1000.00";col(15);
176:         loop=1;
177:     }
178:     else loop=0;
179:     }while(loop==1);
180:
181:     col(121);cout<<"\n\n\t                    ";
182:     cout<<"\n\t  ACCOUNT CREATED!  ";
183:     cout<<"\n\t                    ";col(15);
184: }
185:
186: void account::show() const
187: {
188:     system("cls");
189:     logoandtime();
190:     col(112);cout<<"\n\tAccount No.: "<<accNo;
191:     col(15);
192:     cout<<"\n\n\tAccount Holder Name: ";
193:     cout<<accName;
194:     cout<<"\n\tType of Account: "<<accType;
195:     cout<<"\n\tRemaining Balance: Php "<<fixed<<setprecision(2)<<accBalance<<endl<<endl;
196:     cout<<"\t";system("pause");
197: }
198:
199:
200: void account::modify(int x)
201: {
202:     system("cls");
203:     logoandtime();
204:
205:     if(x==1){
206:
207:     col(112); cout<<"\n\n\tModify Account Holder Name: "; col(15);
208:
209:     int x=0,loop=0;
210:     char tempaccName[50];
211:     do
212:     {
213:         cin.ignore();
214:         cin.getline(tempaccName,50);
215:         x=0;
216:             for(int i=0 ; i < strlen(tempaccName); i++) {
217:                 if(!isalpha(tempaccName[i]) )   {
218:                     loop = 1;
219:                     x++;
220:                     if(isspace(tempaccName[i])){
```

```
221:                     x--;
222:                     }
223:                     if(ispunct(tempaccName[i])){
224:                     x--;
225:                     }
226:                 }
227:             }
228:
229:     if(x==0){
230:         strcpy(accName, tempaccName);
231:         loop = 0;
232:     }
233:     else {
234:         col(12);cout <<"\tEnter a Valid Name: ";
235:     }
236:
237:
238:     }while(loop == 1);
239:     col(112); cout<<"\n\tModify Account PIN: "; col(15);
240:     cin >> accPin;}
241:     else if(x==2){
242:     col(112); cout<<"\n\n\tModify Account PIN: "; col(15);
243:     cin >> accPin;}
244:
245: }
246:
247: void account::deposit(int x)
248: {
249:     accBalance+=x;
250: }
251:
252: void account::withdraw(int x)
253: {
254:     accBalance-=x;
255: }
256:
257: void account::accDetails(int x) const
258: {
259:     pos(10,x);cout<<accNo;
260:     pos(25,x);cout<<accName;
261:     pos(51,x);cout<<accType;
262:     pos(59,x);cout<<"Php "<<fixed<<setprecision(2)<<accBalance;
263: }
264:
265: int account::retaccNo() const
266: {
267:     return accNo;
268: }
269:
270: int account::retaccBalance() const
271: {
272:     return accBalance;
273: }
274:
275: char account::retaccType() const
```

```
{
    return accType;
}

int account::retaccPin() const
{
    return accPin;
}

void writeAccount()
{


    ofstream outFile;
    outFile.open("accdata.dat",ios::binary|ios::app);
    ac.createAccount();
    outFile.write(reinterpret_cast<char *> (&ac), sizeof(account));
    outFile.close();
}

//Functions to read specific record from file

void displayAccount(int n)
{
    logoandtime();

    bool flag=false;
    ifstream inFile;
    inFile.open("accdata.dat",ios::binary);
    if(!inFile)
    {
        col(12);pos(2,12);cout<<"\n\tNo records found!";col(15);
        return;
    }
    system("cls");
    cout<<"\n\tBALANCE DETAILS\n";

        while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
    {
        if(ac.retaccNo()==n)
        {
            ac.show();
            flag=true;
        }
    }
    inFile.close();
    if(flag==false)
    {
        col(124);cout<<"\n\n\t                              ";
                 cout<<"\n\t  ACCOUNT DOES NOT EXIST!   ";
                 cout<<"\n\t                              ";col(15);
    }
}

//Function to modify record of file
```

```
331: void modifyAccount(int n,int m)
332: {
333:     bool found=false;
334:
335:     fstream File;
336:     File.open("accdata.dat",ios::binary|ios::in|ios::out);
337:     if(!File)
338:     {
339:         col(12);pos(2,12);cout<<"\n\tNo records found!";col(15);
340:         return;
341:     }
342:     while(!File.eof() && found==false)
343:     {
344:         File.read(reinterpret_cast<char *> (&ac), sizeof(account));
345:         if(ac.retaccNo()==n)
346:         {
347:             ac.show();
348:             ac.modify(m);
349:             int pos=(-1)*static_cast<int>(sizeof(account));
350:             File.seekp(pos,ios::cur);
351:             File.write(reinterpret_cast<char *> (&ac), sizeof(account));
352:             col(121);cout<<"\n\n\t                    ";
353:             cout<<"\n\t  RECORD UPDATED!  ";
354:             cout<<"\n\t                   ";col(15);
355:             found=true;
356:           }
357:     }
358:     File.close();
359:     if(found==false){
360:         col(124);cout<<"\n\n\t                           ";
361:                  cout<<"\n\t  ACCOUNT DOES NOT EXIST!  ";
362:                  cout<<"\n\t                           ";col(15);}
363: }
364:
365: //Function to delete record of file
366: void deleteAccount(int n)
367: {
368:
369:     ///////////////
370:     bool found=false;
371:     fstream File;
372:     File.open("accdata.dat",ios::binary|ios::in|ios::out);
373:     while(!File.eof() && found==false)
374:     {
375:         File.read(reinterpret_cast<char *> (&ac), sizeof(account));
376:         if(ac.retaccNo()==n)
377:         {
378:         ac.show();
379:         found=true;}
380:     }
381:     File.close();
382:     if(found==false)
383:     {
384:         col(124);cout<<"\n\n\t                           ";
385:                  cout<<"\n\t  ACCOUNT DOES NOT EXIST!  ";
```

```
386:                 cout<<"\n\t                              ";col(15);
387:             }
388:     else
389:     {
390:         col(121);cout<<"\n\n\t                    ";
391:                 cout<<"\n\t  ACCOUNT CLOSED!  ";
392:                 cout<<"\n\t                    ";col(15);
393:     }
394:         //////////////
395:     ifstream inFile;
396:     ofstream outFile;
397:     inFile.open("accdata.dat",ios::binary);
398:     if(!inFile)
399:     {
400:         cout<<"File could not be open !! Press any Key...";
401:         return;
402:     }
403:     outFile.open("Temp.dat",ios::binary);
404:     inFile.seekg(0,ios::beg);
405:     while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
406:     {
407:         if(ac.retaccNo()!=n)
408:         {
409:             outFile.write(reinterpret_cast<char *> (&ac), sizeof(account));
410:         }
411:     }
412:     inFile.close();
413:     outFile.close();
414:     remove("accdata.dat");
415:     rename("Temp.dat","accdata.dat");
416: }
417:
418: //Function to display all accounts
419: void dispAll()
420: {
421:
422:     ifstream inFile;
423:     inFile.open("accdata.dat",ios::binary);
424:     if(!inFile)
425:     {
426:         pos(2,12);cout<<"File could not be open !! Press any Key...";
427:         return;
428:     }
429:     col(112);
430:     cout<<"\n\n\n\t=============================================================\n";
431:     col(124);cout<<"\t Account No.            NAME              Type      Balance     \n";
432:     col(112);cout<<"\t=============================================================\n";
433:     col(15);
434:     int i=11;
435:     while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account))) {
436:         ac.accDetails(i);
437:         i++;
438:     }
439:     inFile.close();
440: }
```

```
441:
442: //Function to deposit and withdraw amounts
443: void depwith(int n, int option)
444: {
445:     int amt;
446:     bool found=false;
447:
448:     fstream File;
449:     File.open("accdata.dat", ios::binary|ios::in|ios::out);
450:     if(!File)
451:     {
452:         col(12);pos(2,12);cout<<"\n\tNo record found!";col(15);
453:         return;
454:     }
455:     while(!File.eof() && found==false)
456:     {
457:         File.read(reinterpret_cast<char *> (&ac), sizeof(account));
458:         if(ac.retaccNo()==n)
459:         {
460:             ac.show();
461:             if(option==1)
462:             {
463:                 col(112);cout<<"\n\n\t DEPOSIT AMOUNT ";col(15);
464:                 cout<<"\n\n\tEnter Amount: ";
465:                 cin>>amt;
466:                 ac.deposit(amt);
467:                 ac.show();
468:                     col(121);cout<<"\n\n\t                    ";
469:                     cout<<"\n\t  RECORD UPDATED!  ";
470:                     cout<<"\n\t                    ";col(15);
471:             }
472:             if(option==2)
473:             {
474:                 col(112);cout<<"\n\n\t WITHDRAW AMOUNT ";col(15);
475:                 cout<<"\n\n\tEnter Amount: ";
476:                 cin>>amt;
477:                 int bal=ac.retaccBalance()-amt;
478:
479:                 if((bal<500 && ac.retaccType()=='S') || (bal<1000 && ac.retaccType()=='C')){
480:
481:                     col(12);cout<<"\n\tInsufficient balance!\n"
482:                                 <<"\tMaintaning Balance for:\n"
483:                                 <<"\tCurrent Account: Php 1000.00\n"
484:                                 <<"\tSavings Account: Php 500.00\n";col(15);
485:                     }
486:                 else{
487:                     ac.withdraw(amt);
488:                     ac.show();
489:                     col(124);cout<<"\n\n\t                    ";
490:                     cout<<"\n\t  RECORD UPDATED!  ";
491:                     cout<<"\n\t                    ";col(15);
492:                 }
493:             }
494:             int pos=(-1)*static_cast<int>(sizeof(ac));
495:             File.seekp(pos,ios::cur);
```

```
496:             File.write(reinterpret_cast<char *> (&ac), sizeof(account));
497:
498:             found=true;
499:            }
500:          }
501:     File.close();
502:     if(found==false){
503:
504:         col(124);cout<<"\n\n\t                     ";
505:             cout<<"\n\t  RECORD NOT FOUND!  ";
506:             cout<<"\n\t                     ";col(15);
507:         }
508: }
509:
510: //Intro Function
511: void welcome()
512: {
513:     col(12);pos(2,3);cout <<currentDateTime();
514:     col(192);pos(50,8);cout << " W E L C O M E  T O ";
515:     int k;
516:     col(205);
517:         for (int j=10; j<15;j++){
518:         pos(45,j);
519:              for (int i=0; i<3; i++){
520:                 cout << " " ;
521:                 Sleep(10);
522:          }
523:         }
524:         pos(48,10); cout<<"    ";
525:         pos(48,14); cout<<"    ";
526:         for (int j=10; j<15;j++){
527:         pos(52,j);
528:              for (int i=0; i<3; i++){
529:                 cout << " " ;
530:                 Sleep(10);
531:          }
532:         }
533:         for (int j=10; j<15;j++){
534:         pos(56,j);
535:              for (int i=0; i<3; i++){
536:                 cout << " " ;
537:                 Sleep(10);
538:          }
539:         }
540:         pos(57,14); cout<<"      ";
541:         col(240);
542:         for (int j=10; j<15;j++){
543:         pos(64,j);
544:              for (int i=0; i<3; i++){
545:                 cout << " " ;
546:                 Sleep(10);
547:          }
548:         }
549:         pos(65,10);cout<<"      ";
550:         for (int j=10; j<12;j++){
```

```
551:         pos(71,j);
552:              for (int i=0; i<3; i++){
553:                 cout << " " ;
554:                 Sleep(10);
555:          }
556:         }
557:         pos(65,12);
558:         cout << "          ";
559:         for (int j=12; j<15;j++){
560:         pos(71,j);
561:              for (int i=0; i<3; i++){
562:                 cout << " " ;
563:                 Sleep(10);
564:          }
565:         }
566:         pos(65,14);
567:         cout << "          ";
568:         col(240);pos(45,16);cout << "O R A N G E  L I O N  B A N K";
569:
570:
571:     col(15);
572:     pos(45,20);system("pause>0");
573: }
574:
575: //Function for Login system
576: int login()
577: {
578:     int opt, num,loop;
579:     string adminID, adminPass;
580:     col(121);
581:     cout << "\n\n\t\t\t [1] Admin Login"
582:          << "\n\t\t\t [2] User Login "
583:          << "\n\n\t\t\t Enter option:  ";
584:     col(112);
585:     cin >> opt;
586:     switch (opt)
587:     {
588:         case 1:
589:         do{
590:             loop=0;
591:             col(121);cout << "\n\t\t\tEnter Admin ID: ";
592:             col(112);cin >> adminID;
593:             col(121);cout << "\t\t\tEnter Admin Password: ";
594:             col(112);cin >> adminPass;
595:
596:             if((adminID=="admin")&&(adminPass=="1234"))
597:             {
598:                 col(121);
599:                 cout<<"\n\t\t\t                    ";
600:                 cout<<"\n\t\t\t   WELCOME ADMIN!   ";
601:                 cout<<"\n\t\t\t                    \n\n\t\t\t";
602:                 col(15);
603:                 system("pause");
604:             }
605:             else
```

```
606:             {
607:                 col(12);
608:                 cout<<"\n\t\t\tIncorrect Admin ID and Password    \n\t\t\t";
609:                 col(15);
610:                 system("pause");
611:                 loop=1;
612: 
613:             }
614: 
615:         }while(loop==1);
616:             break;
617:         case 2:
618:             col(112);cout << "\t\t\tEnter Account No.: ";
619:             cin >> num;
620:             usernum = num;
621:             col(15);
622:             matchAccount(num);
623:             break;
624:         default: col(12);cout <<"\n\t\t\tInvalid Option! Goodbye!";col(15);
625:                     return 0; system("pause");col(15);
626:     }
627:     return opt;
628: }
629: 
630: void matchAccount(int n)
631: {int loop;
632:     do
633:     {
634:     loop = 1;
635:     int pin;
636:     bool found1=false,found2=false;
637:     fstream File;
638:     File.open("accdata.dat",ios::binary|ios::in|ios::out);
639: 
640:     while(!File.eof() && found1==false)
641:     {
642:         File.read(reinterpret_cast<char *> (&ac), sizeof(account));
643:         if(ac.retaccNo()==n)
644:         {found1=true;
645:         }
646: 
647:     }
648:     File.close();
649:     if(found1==true)
650:     {
651:         col(112);cout <<"\t\t\tEnter Account PIN: ";
652:         cin>>pin;col(15);
653: 
654: 
655:             while(!File.eof() && found2==false)
656:             {
657: 
658:             File.read(reinterpret_cast<char *> (&ac), sizeof(account));
659:             if(ac.retaccPin()==pin)
660:             {
```

```
661:
662:             found2=true;}
663:             }
664:             File.close();
665:             if((found1==true)&&(found2==true)){
666:
667:             col(121);
668:                 cout<<"\n\t\t\t                    ";
669:                 cout<<"\n\t\t\t   WELCOME USER!    ";
670:                 cout<<"\n\t\t\t                    \n\n\t\t\t";
671:                 col(15);
672:             system("pause");
673:             loop = 0;
674:         }
675:             else {
676:                 col(12);
677:                 cout<<"\t\t\tIncorrect PIN!\n\t\t\t";
678:                 col(15);
679:                 system("pause");
680:             }
681:     }
682:     else
683:     {
684:         col(12);cout <<"\n\t\t\tAccount does not exist!";
685:         col(112);cout << "\n\n\t\t\tEnter Account No.: ";
686:         cin >> n;col(15);
687:     }
688:
689:
690: }while(loop ==1);
691:
692: }
693:
694: /*******************************************************/
695: //          MAIN PROGRAM                               //
696: /*******************************************************/
697:
698: int main()
699: {
700:     int user;
701:     welcome();
702:     user = login();
703:     string ch;
704:     int num;
705:     if(user==1)
706: {
707:     do
708:     {
709:
710:         system("cls");
711:         logoandtime();
712:         col(112);
713:         pos(39,5);cout<<"               ";
714:         pos(39,6);cout<<"   MAIN MENU   ";
715:         pos(39,7);cout<<"               ";
```

```
716:         pos(6,10);cout<<"                          ";
717:         pos(6,11);cout<<"   [1] CREATE ACCOUNT     ";
718:         pos(6,12);cout<<"                          ";
719:         pos(6,14);cout<<"                          ";
720:         pos(6,15);cout<<"   [2] DEPOSIT            ";
721:         pos(6,16);cout<<"                          ";
722:         pos(6,18);cout<<"                          ";
723:         pos(6,19);cout<<"   [3] WITHDRAW           ";
724:         pos(6,20);cout<<"                          ";
725:         pos(6,22);cout<<"                          ";
726:         pos(6,23);cout<<"   [4] BALANCE INQUIRY    ";
727:         pos(6,24);cout<<"                          ";
728:         pos(35,10);cout<<"                          ";
729:         pos(35,11);cout<<"   [5] VIEW ALL ACCOUNTS ";
730:         pos(35,12);cout<<"                          ";
731:         pos(35,14);cout<<"                          ";
732:         pos(35,15);cout<<"   [6] CLOSE ACCOUNT      ";
733:         pos(35,16);cout<<"                          ";
734:         pos(35,18);cout<<"                          ";
735:         pos(35,19);cout<<"   [7] MODIFY ACCOUNT     ";
736:         pos(35,20);cout<<"                          ";
737:         pos(35,22);cout<<"                          ";
738:         pos(35,23);cout<<"   [8] EXIT               ";
739:         pos(35,24);cout<<"                          ";
740:         col(124);
741:         pos(63,17);cout<<"Enter Option (1-8): ";
742:         col(112);cin >>ch; col(15);
743:
744:         system("cls");
745:
746:         if (ch=="1")
747:             writeAccount();
748:
749:         else if(ch=="2"){
750:             logoandtime();
751:             cout<<"\n\n\tEnter Account No. : "; cin>>num;
752:             depwith(num, 1);
753:         }
754:         else if(ch=="3"){
755:             logoandtime();
756:             cout<<"\n\n\tEnter Account No. : "; cin>>num;
757:             depwith(num, 2);
758:             }
759:         else if(ch=="4"){
760:             logoandtime();
761:             cout<<"\n\n\tEnter Account No. : "; cin>>num;
762:             displayAccount(num);
763:         }
764:         else if(ch=="5"){
765:         logoandtime();
766:             dispAll();
767:         }
768:         else if(ch=="6"){
769:         logoandtime();
770:             cout<<"\n\n\tEnter Account No. : "; cin>>num;
```

```
771:             deleteAccount(num);
772:         }
773:          else if(ch=="7"){
774:         logoandtime();
775:             cout<<"\n\n\tEnter Account No. : "; cin>>num;
776:             modifyAccount(num,1);
777:         }
778:          else if(ch=="8"){
779:             logoandtime();
780:             col(15);
781:             cout<<"\n\n\tThank you! Have a nice day!\n";col(15);
782:         }
783:          else {
784:          logoandtime();
785:                 col(12); cout<<"\n\n\tINVALID CHOICE\n";col(15);
786:         }
787:         cin.ignore();
788:         cin.get();
789:     }while(ch!="8");
790: }
791:     else if(user == 2)
792:     {
793:             do
794:     {
795:
796:         system("cls");
797:         logoandtime();
798:         col(112);
799:         pos(39,5);cout<<"                ";
800:         pos(39,6);cout<<"   MAIN MENU    ";
801:         pos(39,7);cout<<"                ";
802:         pos(6,10);cout<<"                              ";
803:         pos(6,11);cout<<"   [1] DEPOSIT                ";
804:         pos(6,12);cout<<"                              ";
805:         pos(6,14);cout<<"                              ";
806:         pos(6,15);cout<<"   [2] WITHDRAW               ";
807:         pos(6,16);cout<<"                              ";
808:         pos(6,18);cout<<"                              ";
809:         pos(6,19);cout<<"   [3] BALANCE INQUIRY        ";
810:         pos(6,20);cout<<"                              ";
811:         pos(6,22);cout<<"                              ";
812:         pos(6,23);cout<<"   [4] CHANGE PIN             ";
813:         pos(6,24);cout<<"                              ";
814:         pos(6,26);cout<<"                              ";
815:         pos(6,27);cout<<"   [5] EXIT                   ";
816:         pos(6,28);cout<<"                              ";
817:         col(124);
818:         pos(39,17);cout<<"Enter Option (1-5): ";
819:         col(112);cin >>ch; col(15);
820:
821:         system("cls");
822:
823:
824:         if(ch=="1"){
825:
```

```
826:             logoandtime();
827:             num=usernum;
828:             depwith(num, 1);
829:         }
830:         else if(ch=="2"){
831:         logoandtime();
832:             num=usernum;
833:             depwith(num, 2);
834:         }
835:         else if(ch=="3"){
836:         logoandtime();
837:             num=usernum;
838:             displayAccount(num);
839:         }
840:         else if(ch=="4"){
841:         logoandtime();
842:             num=usernum;
843:             modifyAccount(num,2);}
844:         else if (ch=="5"){
845:         logoandtime();
846:             col(15);
847:             cout<<"\n\n\tThank you! Have a nice day!\n";
848:             ch = '8';
849:         }
850:         else {
851:         logoandtime();
852:                 col(12); cout<<"\n\n\tINVALID CHOICE\n";col(15);
853:         }
854:         cin.ignore();
855:         cin.get();
856:     }while(ch!="8");
857:     }
858:
859:
860: }
861: ////////////////////////////END OF PROGRAM/////////////////////////////////
862: //         YEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEESSSS
```