

|  |
| --- |
| COMPUTER SCIENCE PROJECT  2018 |
|  |
| AKSHAYA RAVIKUMAR, anusha nath roy, meena nagrajan XII-D |

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NO |
| 1. | ACKNOWLEDGEMENT |  |
| 2. | AIM |  |
| 3. | SOFTWARE AND HARDWARE SPECIFICATIONS |  |
| 4. | HEADER FILES USED |  |
| 5. | BRIEF EXPLANATION |  |
| 6. | SOURCE CODE |  |
| 7. | SAMPLE OUTPUT |  |
| 8. | CONCLUSION |  |
| 9. | BIBLIOGRAPHY |  |

*CONTENTS*

*ACKNOWLEDGEMENT*

I would like to express a deep sense of thanks to my computer science teacher Mrs. Juhi Sayyed for guiding me immensely through the course of the project. Her advice and constant motivation have been responsible for the successful completion of this project.

I also thank my project partner for her immense support without whom this project would have never been possible.

I also thank my parents for their motivation and support. I also thank the Almighty for giving me the strength, energy and patience for the completion of this project.

*AIM*

To create a program for e-banking program in C++ using data file handling concepts.

*SOFTWARE AND HARDWARE SPECIFICATIONS*

SOFTWARE:

BASIC REQUIREMENTS: TURBO c++

HARDWARE:

Cd rom Speed : 52 x

Processor: Pentium 4

Hard disk capacity: 20gb

Memory: 512 mb ram

Aim

Explain in brief about project

*HEADER FILES*

1.#include<fstream.h> - for file handling, cin and cout.

2.#include<string.h> - for string handling

3.#include<ctype.h> - for character handling

4.#include<iomanip.h> - for parametric manipulators

5.#include<stdio.h> - for standard I/O operations

6.#include<math.h> - for mathematical operations

7.#include<conio.h> - for clrscr() and getch() functions

*BRIEF EXPLANATION*

1-To search based on type in the bank. – search\_type()

2-To display the details of your account.-display()

3-To search based on account number.-open\_choice

4-To append an account by adding new records.-append()

5-To deposit amount in your account.-deposit()

6-To withdraw amount from your account.-withdraw()

7-To take a loan from the bank.-loan()

*SOURCE CODE*

#include<conio.h>

#include<fstream.h>

#include<string.h>

#include<ctype.h>

#include<iomanip.h>

#include<stdio.h>

#include<math.h>

class account

{

int acno;

char name[50];

int deposit;

char type[3];

public:

void create\_account();

void show\_account();

void f\_deposit(int acno);

void draw(int acno);

int return\_acno();

int return\_deposit();

char\* return\_type();

};

void account::create\_account()

{

cout<<"\n Enter the account number:";

cin>>acno;

cout<<"Enter the name of the account holder: ";

gets(name);

cout<<"Enter the type of the account (C/S/FD): ";

gets(type);

cout<<"Enter the initial amout of deposit: ";

cin>>deposit;

cout<<"YOUR ACCOUNT HAS BEEN CREATED."<<"\n";

}

void account::show\_account()

{

cout<<"\n Account Number: "<<acno;

cout<<"\n Name of the Account Holder: ";

cout<<name;

cout<<"\n Type of Account: "<<type;

cout<<"\n Balance Account: "<<deposit;

}

void account::f\_deposit(int y)

{

deposit=deposit+y;

}

void account::draw(int y)

{

deposit=deposit-y;

}

int account::return\_acno()

{

return acno;

}

int account::return\_deposit()

{

return deposit;

}

char\* account::return\_type()

{

return type;

}

void Intro()

{

cout<<"WELCOME TO BYTES BANK."<<endl;

}

//create a new file

void create()

{

int i,n;

account a;

ofstream fout;

cout<<"How many records you want?";

cin>>n;

fout.open("bank.dat",ios::out|ios::binary);

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

{

a.create\_account();

fout.write((char\*)&a,sizeof(a));

}

fout.close();

}

//read file

void display()

{

account a;

ifstream fin;

fin.open("bank.dat",ios::in|ios::binary);

fin.seekg(0);

while(fin.read((char\*)&a,sizeof(a)))

{

a.show\_account();

}

fin.close();

}

//search based on account type

void search\_type()

{

account a;

char choice\_type[3],flag=0;

ifstream fin;

fin.open("bank.dat",ios::in|ios::binary);

cout<<"Enter account number of your choice: ";

cin>>choice\_type;

while(fin.read((char\*)&a,sizeof(a)))

{

if(strcmpi(a.return\_type(),choice\_type)==0)

{

a.show\_account();

flag=1;

}

}

fin.close();

if (flag==0)

{

cout<<"Sorry! Account not found! Please check again!";

}

}

//search based on account number

void open\_choice()

{

account a;

int choice\_acno,flag=0;

ifstream fin;

fin.open("bank.dat",ios::in|ios::binary);

cout<<"Enter TYPE of your choice: ";

cin>>choice\_acno;

while(fin.read((char\*)&a,sizeof(a)))

{

if(a.return\_acno()==choice\_acno)

{

a.show\_account();

flag=1;

}

}

fin.close();

if (flag==0)

{

cout<<"Sorry! Account not found! Please check again!";

}

}

//to append by adding new records

void append()

{

int i,n;

account a;

ofstream fout;

cout<<"How many accounts you want to add?";

cin>>n;

fout. open("bank.dat",ios::ate|ios::binary);

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

{

a.create\_account();

fout.write((char\*)&a,sizeof(a));

}

fout.close();

}

void deposit()

{

account a ;

int accnum;

int depamount;

int flag=0;

fstream f;

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

f.seekg(0);

cout<<"Enter account number: ";

cin>>accnum;

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

{

if(accnum==a.return\_acno())

{

flag=1;

cout<<"Enter the amount to be deposited: ";

cin>>depamount;

a.f\_deposit(depamount);

f.seekp(f.tellg()-sizeof(a),ios::cur);

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

}

}

if (flag==0)

{ cout<<"ERROR! ACCOUNT NOT FOUND.";}

a.show\_account();

f.close();

}

void withdraw()

{

account a;

int accnum;

int wamount;

int flag=0;

fstream f;

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

f.seekg(0);

cout<<"Enter the Account Number: ";

cin>>accnum;

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

{

if(accnum==a.return\_acno())

{flag=1;

cout<<"Enter the amount to be withdrawn: ";

cin>>wamount;

if(strcmpi(a.return\_type(),"s")==0)

{

cout<<"Sorry! Cannot withdraw from savings amount.";

}

else if((a.return\_deposit()-wamount)<10)

{

cout<<"Sorry! Enough money not available in your account.";}

else

{

a.draw(wamount);

f.seekp(f.tellg()-sizeof(a));

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

flag=1;

a.show\_account();

}

}

}

if (flag==0)

cout<<"Error! Account not found!";

f.close();

}

float calculate(int P,int T)

{

float tot=P\*pow(1.15,T-1);

return tot;

}

void loan()

{

account a;

char choice;

int accnum,lamount,years;

fstream f;

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

f.seekg(0);

cout<<"Enter your Account Number: ";

cin>>accnum;

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

{

if(accnum==a.return\_acno())

{

f.seekp(f.tellg()-sizeof(a));

cout<<"Enter the amount you want a loan: ";

cin>>lamount;

cout<<"\n Enter tentative number of years after which you will be able to pay back the loan: ";

cin>>years;

cout<<"\n The interest rate is 15% compounded annually.";

cout<<"\n Amount to be paid after the said time: "<<calculate(lamount,years);

cout<<"\n Do you agree to this?(Y\N)";

cin>>choice;

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

{

a.f\_deposit(lamount);

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

cout<<"The money has been deposited to your account.";

}

else

{

cout<<"Error! Account not found!";

}

}

}

f.close();

}

void main()

{

clrscr();

Intro();

create();

int ch;

char choice;

do

{

cout<<"\n MENU: ";

cout<<"Enter 1-To search based on type in the bank."<<"\n";

cout<<"Enter 2-To display the details of your account."<<"\n";

cout<<"Enter 3-To search based on account number."<<"\n";

cout<<"Enter 4-To append an account by adding new records."<<"\n";

cout<<"Enter 5-To deposit amount in your account."<<"\n";

cout<<"Enter 6-To withdraw amount from your account."<<"\n";

cout<<"Enter 7-To take a loan from the bank."<<"\n";

cout<<"Enter Choice: ";

cin>>ch;

switch(ch)

{

case 1:search\_type();

break;

case 2:display();

break;

case 3:open\_choice();

break;

case 4:append();

break;

case 5:deposit();

break;

case 6:withdraw();

break;

case 7:loan();

break;

default: cout<<"Sorry! Please enter correct choice.";

}

cout<<"\n Do you want to continue?(Y/N)"<<"\n";

cin>>choice;

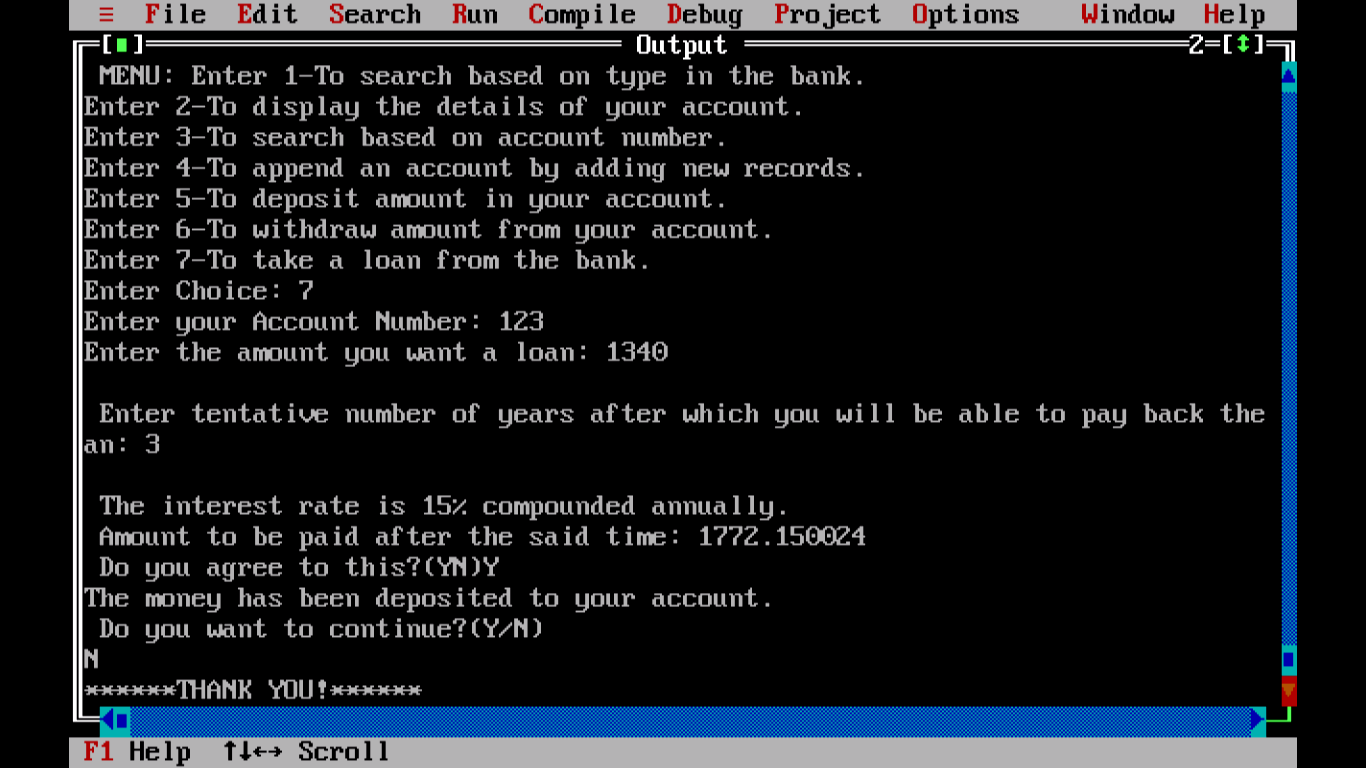
}while(choice=='y'||choice=='Y');

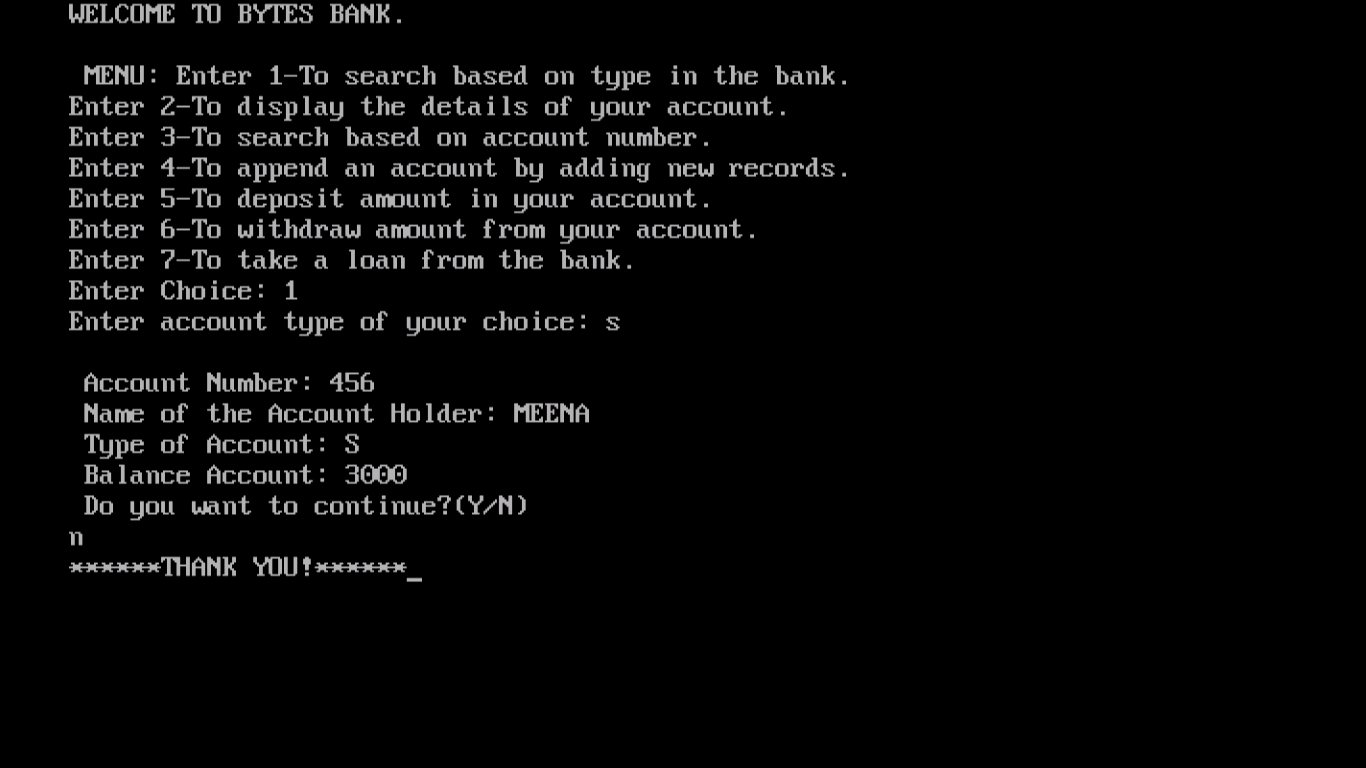
cout<<"\*\*\*\*\*\*THANK YOU!\*\*\*\*\*\*";

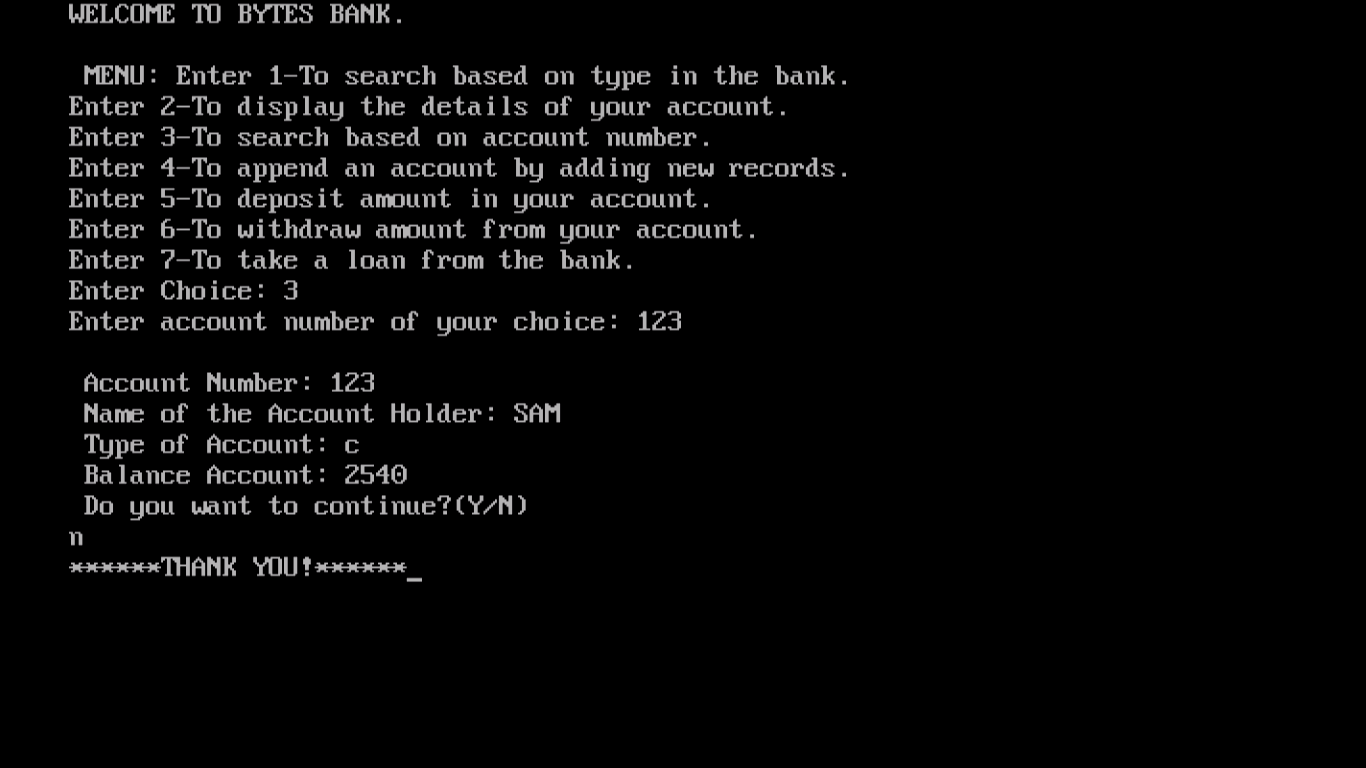
getch();

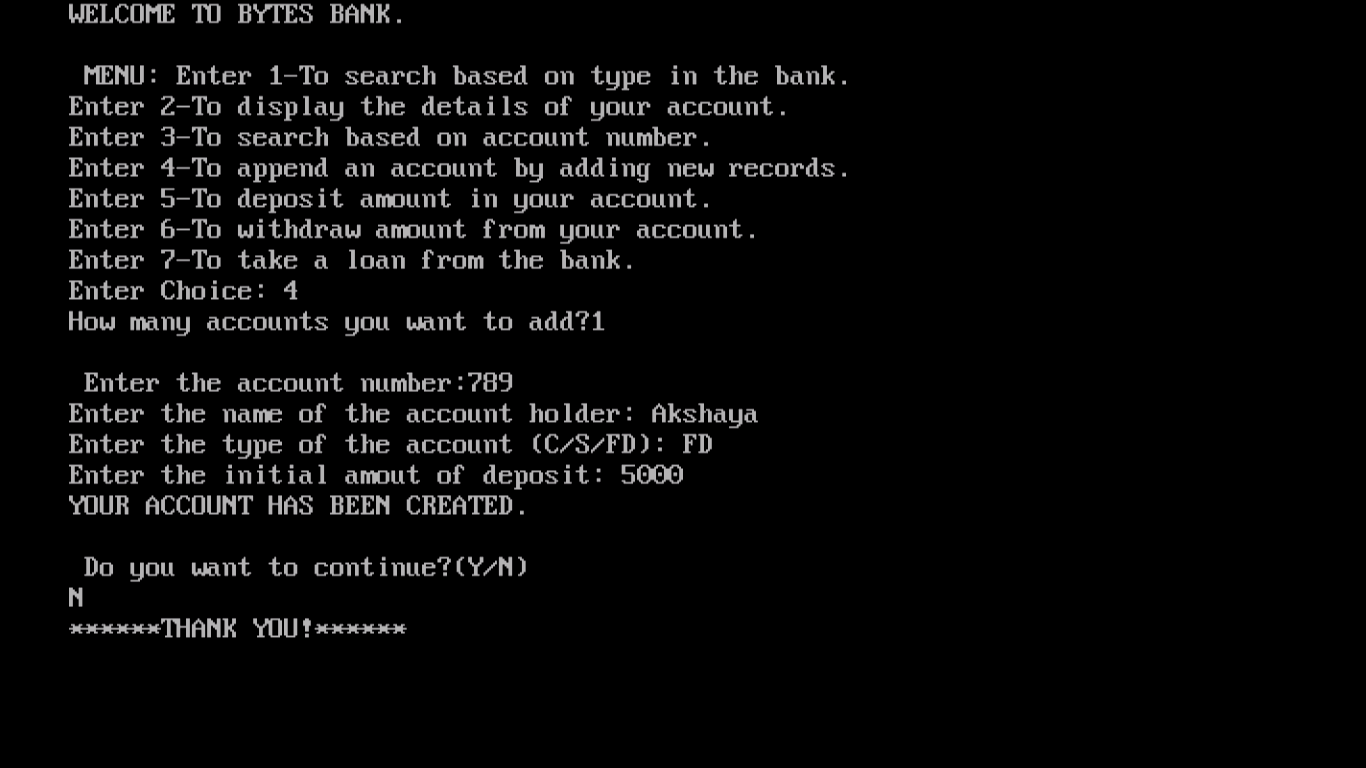
}

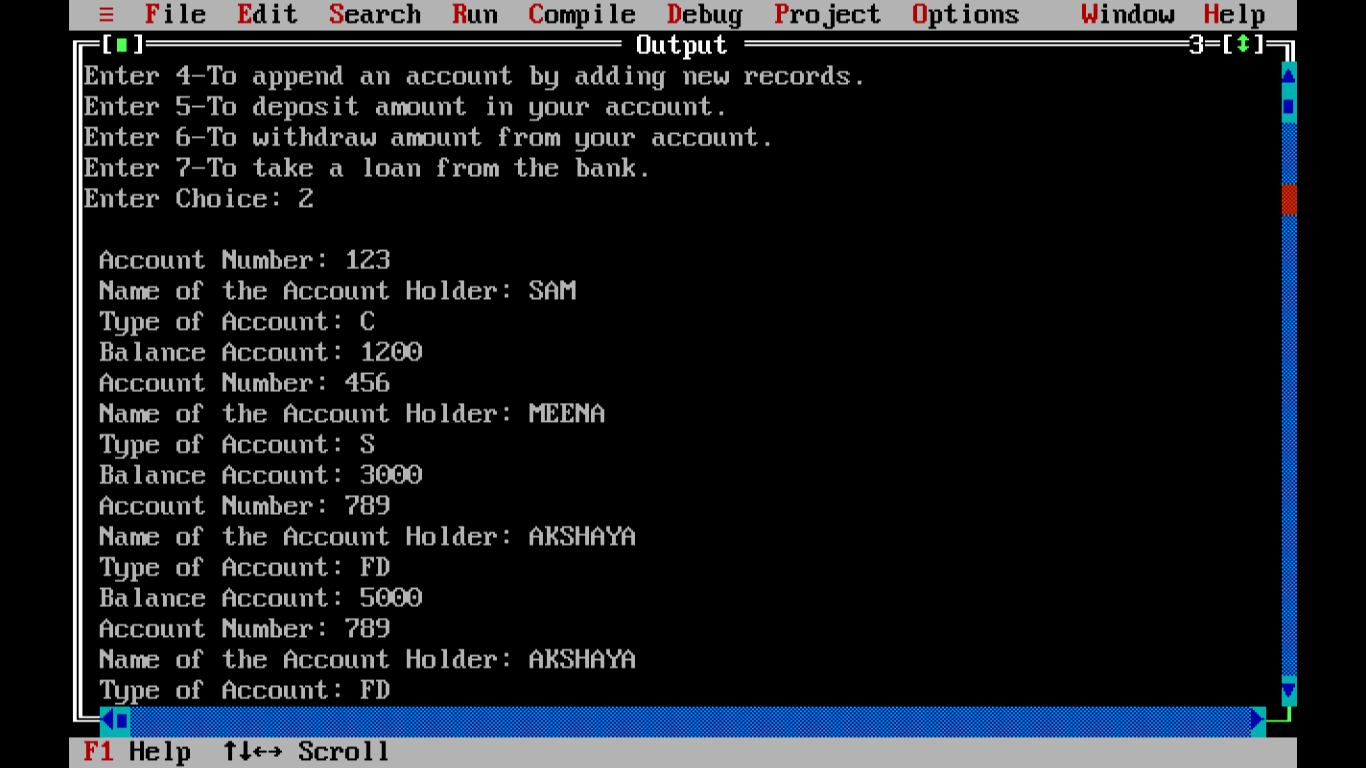
*SAMPLE OUTPUT*

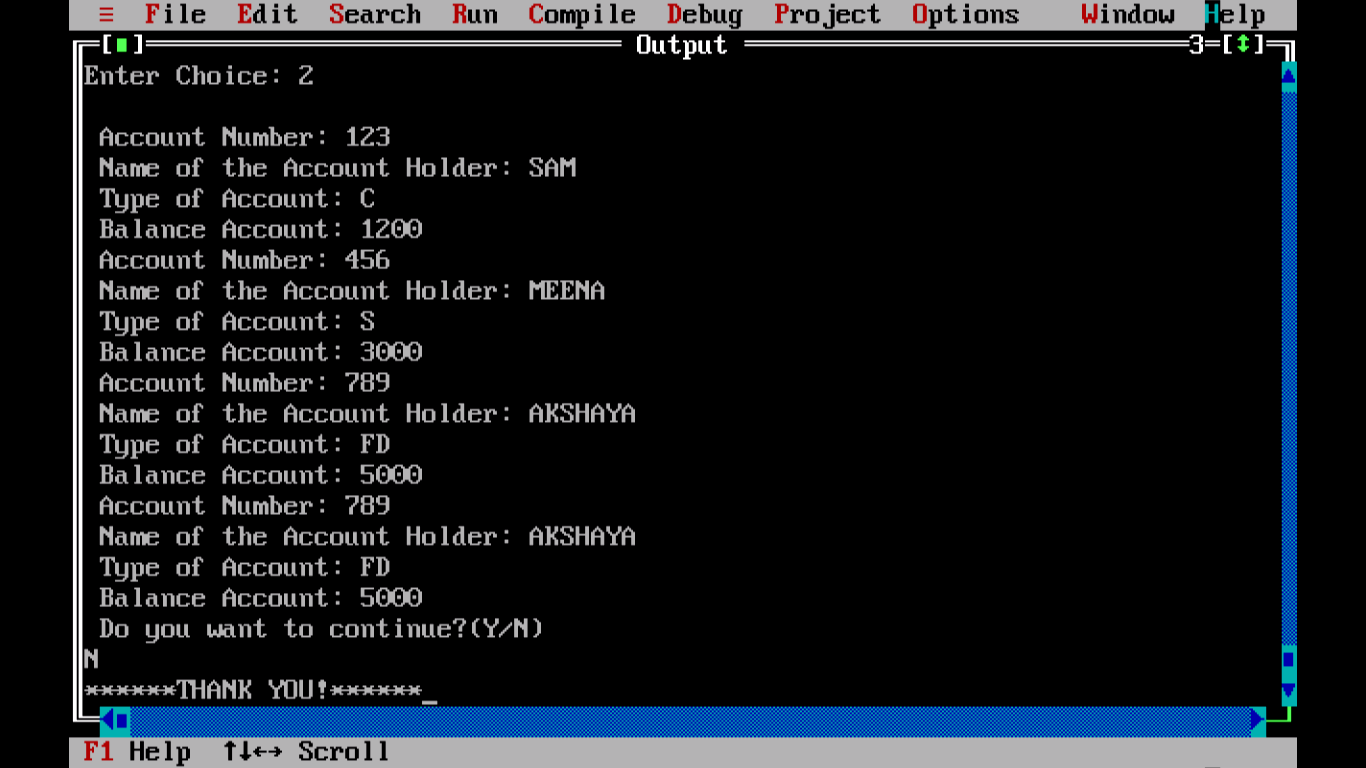
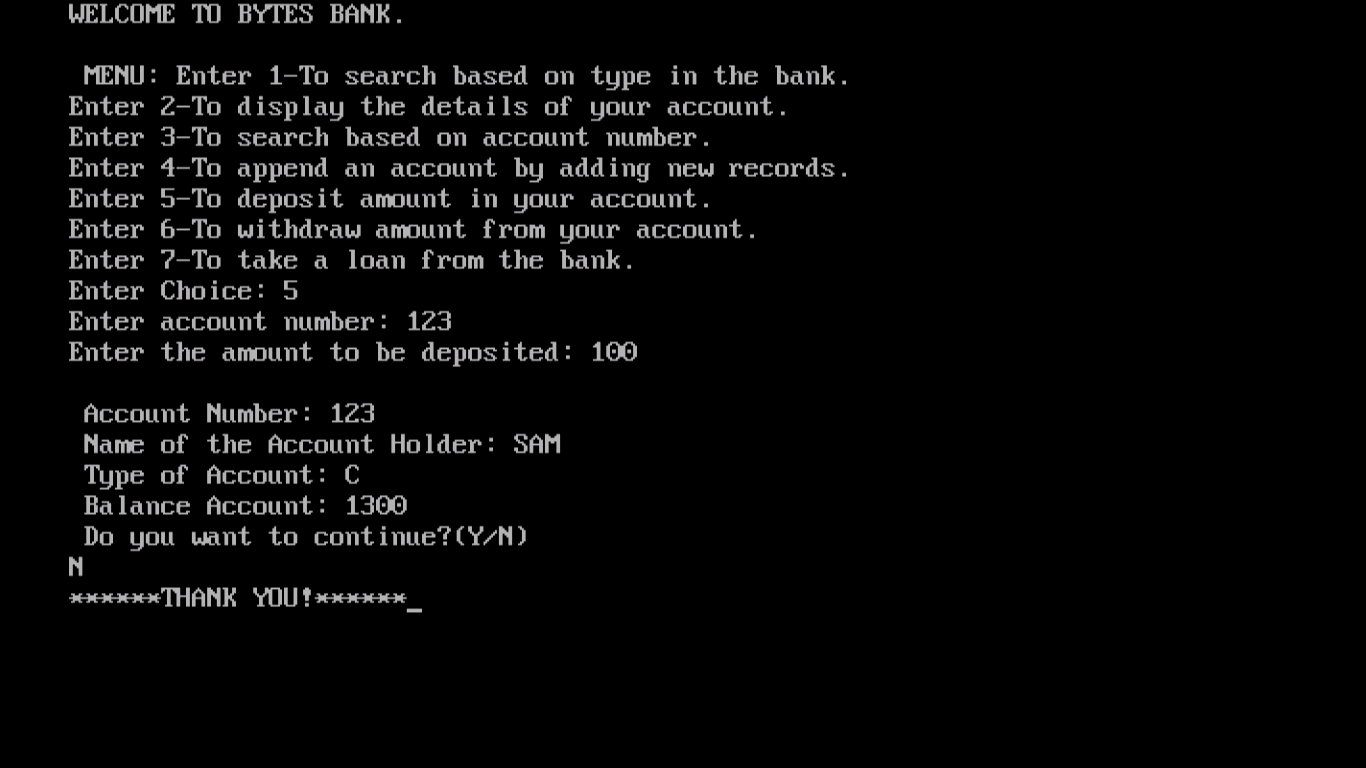












*CONCLUSION*

Program executed successfully!