Banking Application

30-11-25

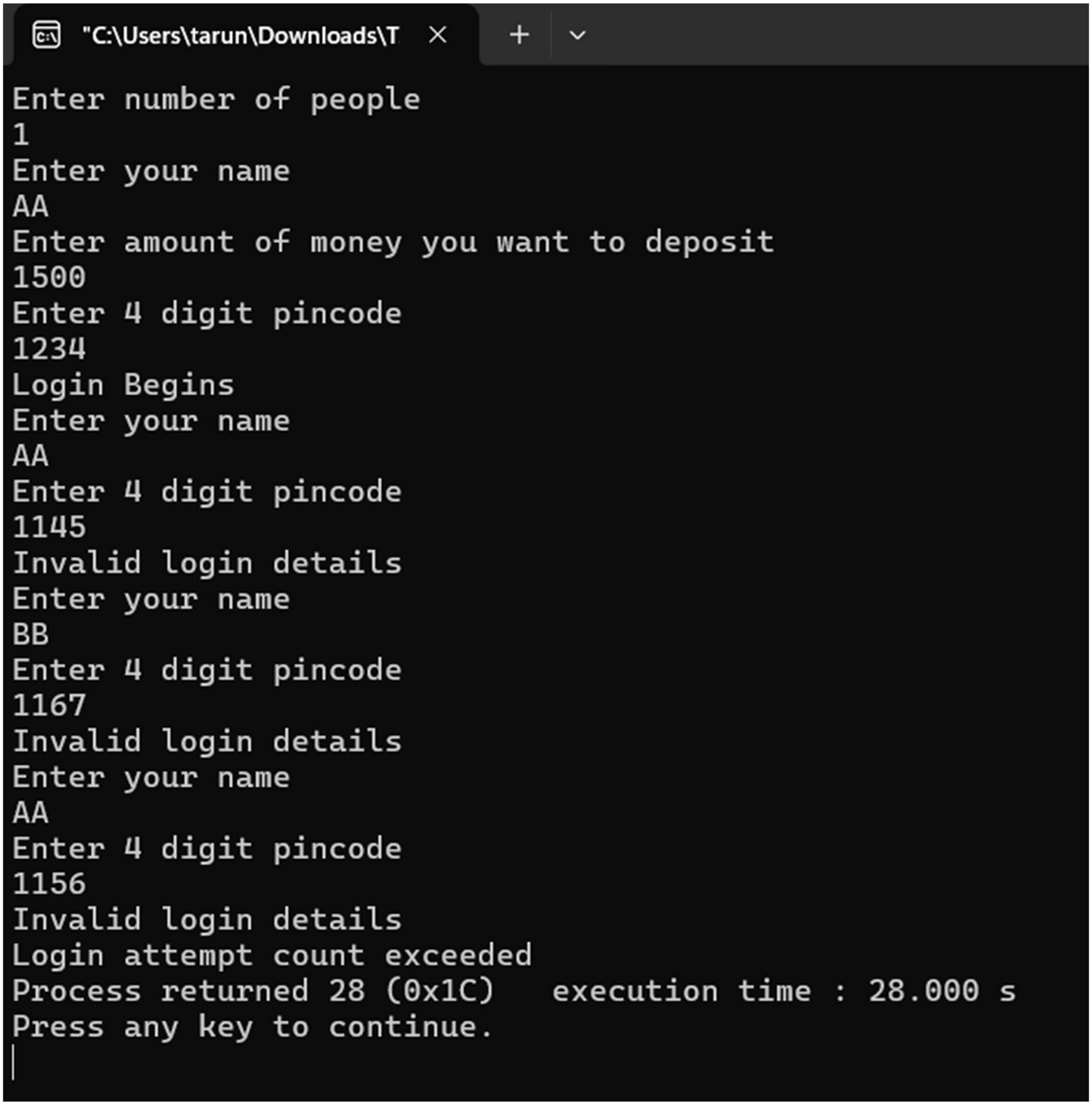
Aim – Write an Application program in C like Library management system/Banking application etc.

Theory –

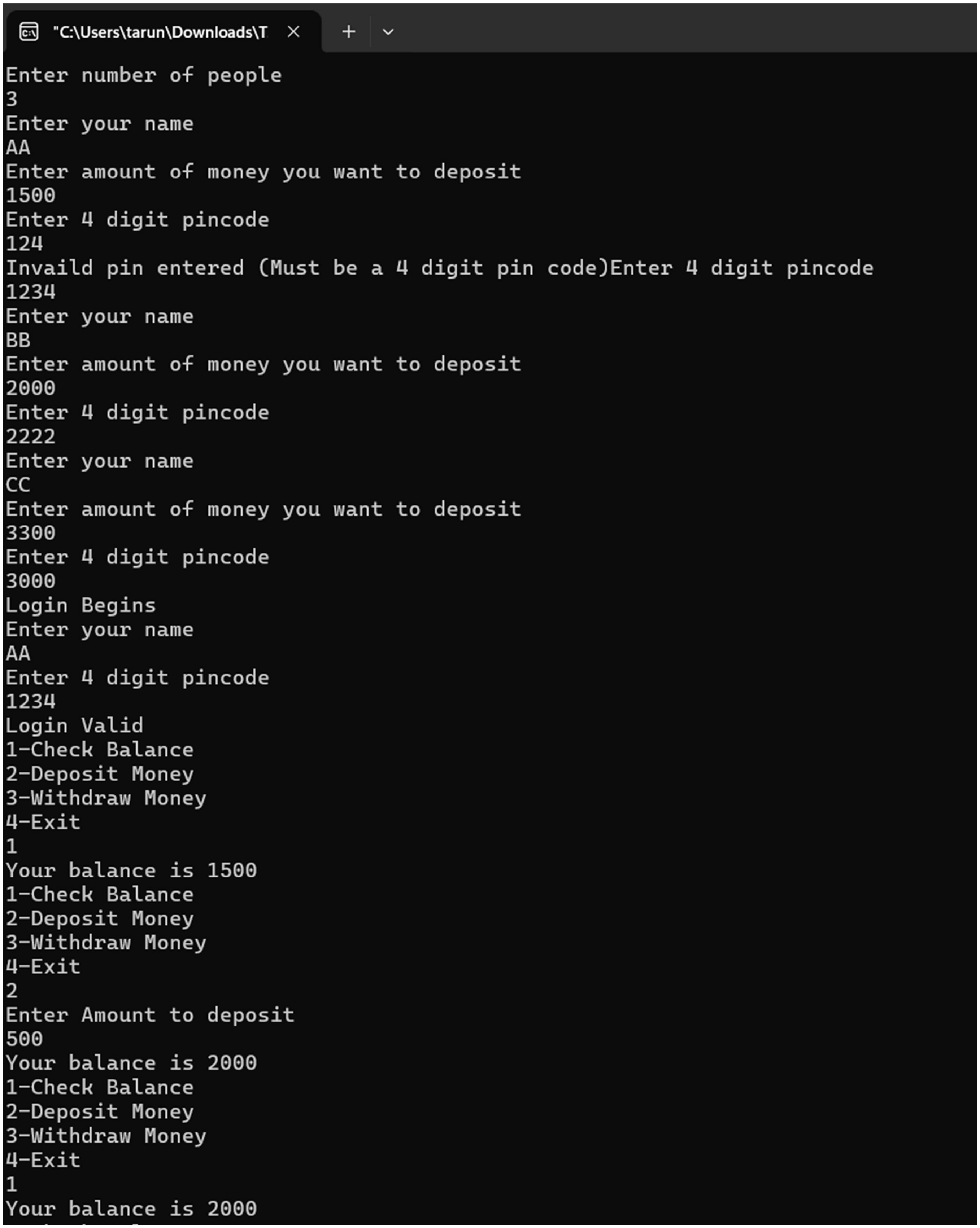
This C program implements a simple banking system for multiple users using structures, arrays, loops, and conditional statements. The struct person defines a blueprint for storing information about each user, including their name, balance, deposited amount, pincode, and transaction details. The program first asks the user for the number of people and initializes an array of struct person accordingly. It then collects each user's name, initial deposit, and 4-digit pincode, validating the pincode using a conditional check.
The program uses a login system to authenticate users, allowing a maximum of three failed attempts. User credentials are verified using strcmp to match names and a comparison operator for the pincode. Once logged in, the user can check balance, deposit money, or withdraw money, using simple arithmetic to update the balance. Conditional statements ensure that withdrawals cannot exceed the available balance. The program also uses loops and goto statements for repeating login attempts and transaction selections.
Dynamic memory is not used; instead, arrays and structures handle multiple users. The program demonstrates key C concepts including arrays, structures, functions (main), conditional statements, loops, string handling, and user input/output, providing a basic but functional banking simulation.

A1.

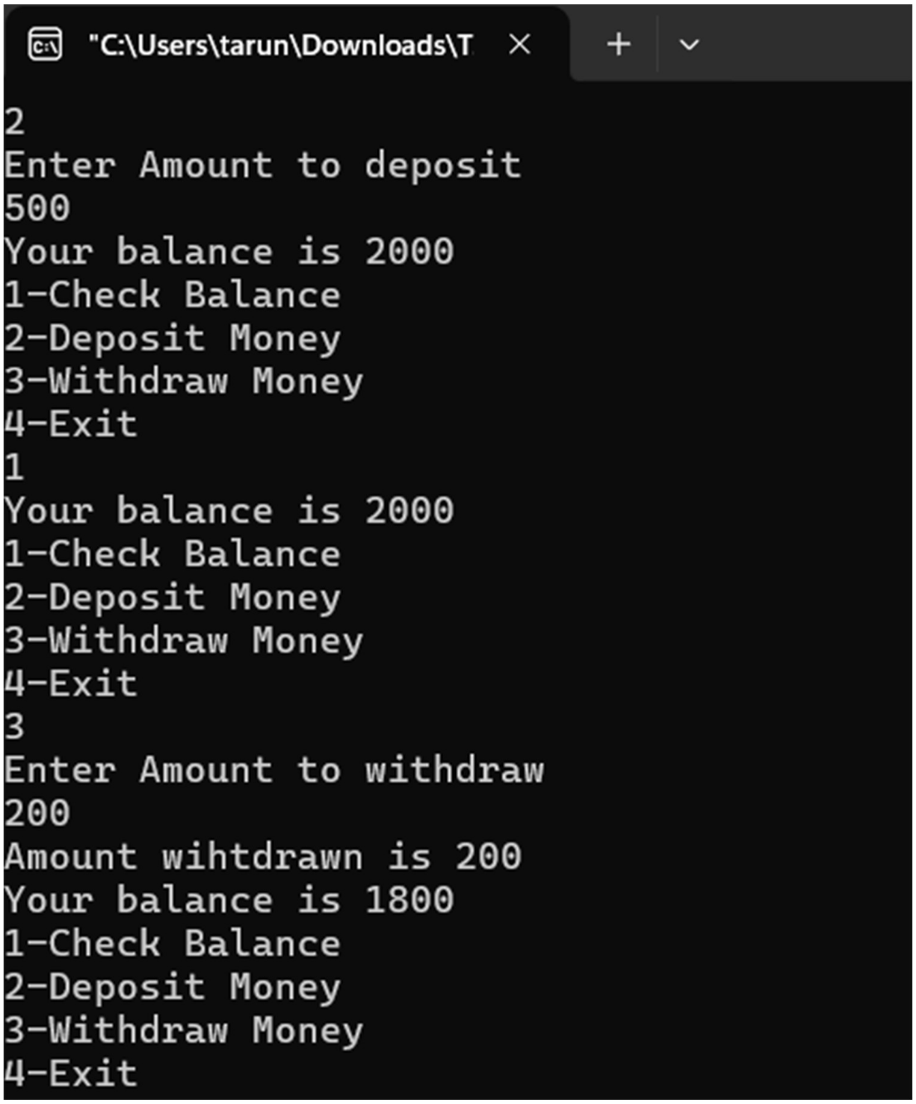
```
#include <stdio.h>
#include <string.h>


struct person {
    char name[20];
    int balance, damount, pin, depositmore, wamount;
};


void main()
{


    int n, tempin, depositmore, wamount;
    char name2[20];
    printf("Enter number of people \n");
    scanf("%d", &n);
    struct person p[n], temp;
    for (int i = 0; i<n; i++){
        printf("Enter your name\n");
        scanf("%s", &p[i].name);
    printf("Enter amount of money you want to deposit\n");
    scanf("%d", &p[i].damount);
    p[i].balance = p[i].damount;

    again:
    printf("Enter 4 digit pincode\n");
    scanf("%d", &p[i].pin);
    if(p[i].pin<999 || p[i].pin>10000){
        printf("Invaild pin entered (Must be a 4 digit pin code)\n");
        goto again;
    }
}
char tempname[20];
int temppin;

printf("Login Begins\n");

int count = 0;
login:
    if (count>=3) {
        printf("Login attempt count exceeded");
        return;
    }
    count ++;
printf("Enter your name\n");
scanf("%s", &tempname);
printf("Enter 4 digit pincode\n");
scanf("%d", &temppin);

int index = -1;
for (int i= 0; i<n;i++){
    if (strcmp(p[i].name, tempname) == 0 && temppin == p[i].pin){
            index = i;
            break;
        }
    }
    if (index == -1){
        printf("Invalid login details\n");
        goto login;
    } else {
        printf("Login Valid\n");
        int choice;
        selection:
        printf("1-Check Balance\n2-Deposit Money\n3-Withdraw Money\n4-Exit\n");
        scanf("%d" , &choice);
        if (choice == 1){
            printf("Your balance is %d\n", p[index].balance);
        } else if (choice == 2) {
            int amt;
            printf("Enter Amount to deposit\n");
            scanf("%d", &amt);
            p[index].balance += amt;
            printf("Your balance is %d\n", p[index].balance);
        } else if (choice == 3) {
            int amt;
            printf("Enter Amount to withdraw\n");
            scanf("%d", &amt);
            if (p[index].balance >= amt){
                p[index].balance -= amt;
            } else {
                printf("Insufficient Balance\n");
                amt = 0;
            }
            printf("Amount wihtdrawn is %d\n", amt);
            printf("Your balance is %d\n", p[index].balance);
        } else if (choice == 4) {
            return;
        } else {
            printf("Invalid choice\n");
        }
            goto selection;
    }
}
```



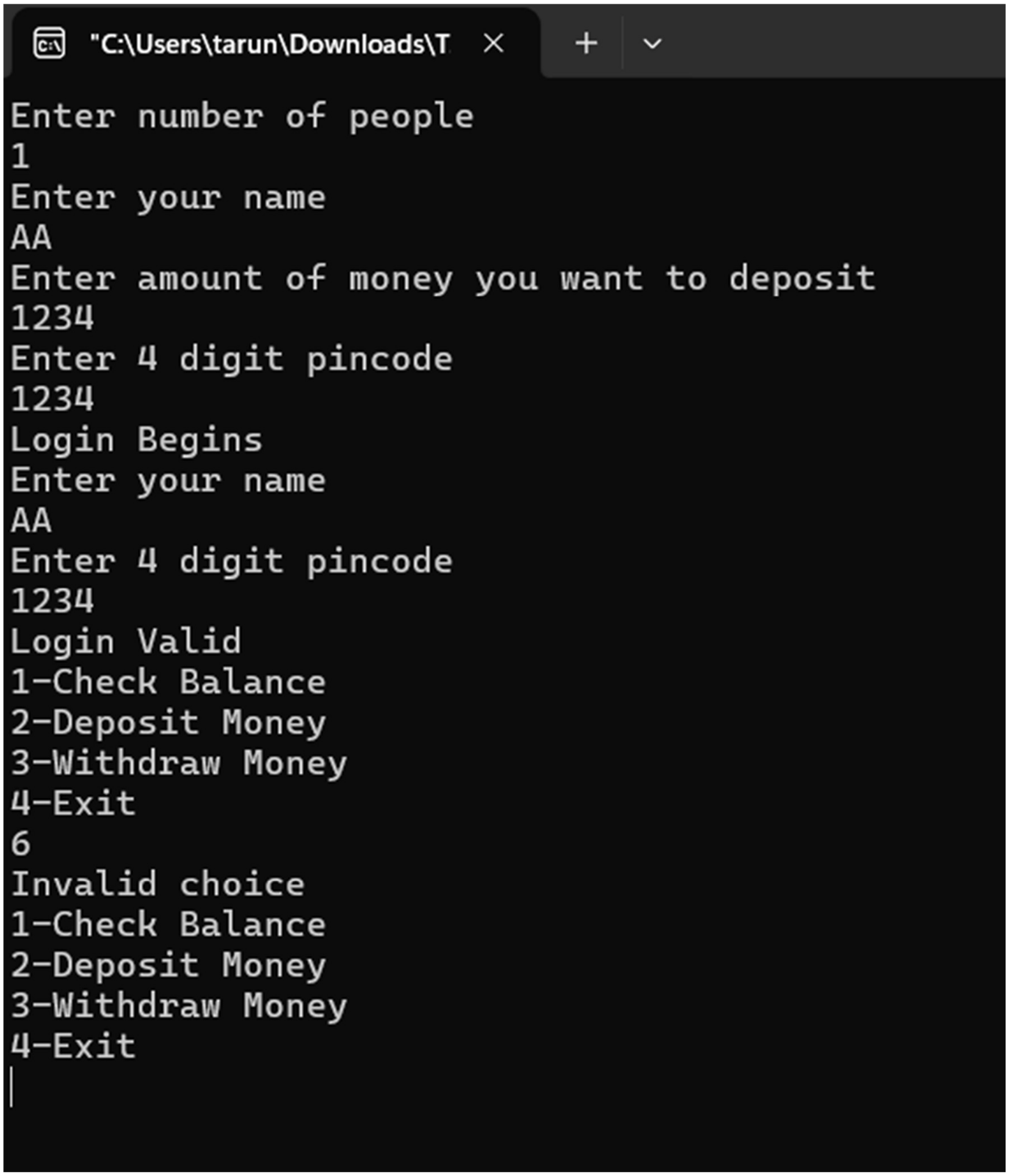
```
Enter number of people
1
Enter your name
AA
Enter amount of money you want to deposit
1500
Enter 4 digit pincode
1234
Login Begins
Enter your name
AA
Enter 4 digit pincode
1145
Invalid login details
Enter your name
BB
Enter 4 digit pincode
1167
Invalid login details
Enter your name
AA
Enter 4 digit pincode
1156
Invalid login details
Login attempt count exceeded
Process returned 28 (0x1C)   execution time : 28.000 s
Press any key to continue.
```



```
Enter number of people
3
Enter your name
AA
Enter amount of money you want to deposit
1500
Enter 4 digit pincode
124
Invaild pin entered (Must be a 4 digit pin code)Enter 4 digit pincode
1234
Enter your name
BB
Enter amount of money you want to deposit
2000
Enter 4 digit pincode
2222
Enter your name
CC
Enter amount of money you want to deposit
3300
Enter 4 digit pincode
3000
Login Begins
Enter your name
AA
Enter 4 digit pincode
1234
Login Valid
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
1
Your balance is 1500
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
2
Enter Amount to deposit
500
Your balance is 2000
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
1
Your balance is 2000
```



```
2
Enter Amount to deposit
500
Your balance is 2000
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
1
Your balance is 2000
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
3
Enter Amount to withdraw
200
Amount wihtdrawn is 200
Your balance is 1800
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
```



```
Enter number of people
1
Enter your name
AA
Enter amount of money you want to deposit
1234
Enter 4 digit pincode
1234
Login Begins
Enter your name
AA
Enter 4 digit pincode
1234
Login Valid
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
6
Invalid choice
1-Check Balance
2-Deposit Money
3-Withdraw Money
4-Exit
```

Conclusion

I learned how to use structures in C to store and manage related data for multiple users. I understood how to take input and validate it, such as enforcing a 4-digit pincode. I practiced arrays, loops, and conditional statements to handle multiple users and transactions efficiently. I learned how to implement a login system with limited attempts and verify user credentials using string comparison. I also gained experience in performing basic banking operations like checking balance, depositing, and withdrawing money, while ensuring balance validation. Overall, this assignment strengthened my understanding of C programming concepts and user interaction logic. I also learnt how to use all the different aspects of c programming in combination with each other.