Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Fall, Year: 2021), B.Sc. in CSE (Day)

Course Title: Structured Programming Lab
Course Code: CSE 104 Section: 212DA

Lab Project Name: BANK MANAGEMENT SYSTEM

Student Details

	Name	ID
1.	Abdul Alim Monshi	212002073

Submission Date : 11 Jan 2022
Course Teacher's Name : Ahmed Iqbal Pritom

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks: …… **Signature: .....**

Comments: ..... **Date: .....**

Table of Contents

Introduction

Originally, C language is developed from two previous languages, BCPL and B. BCPL which were developed in 1967 by Martin Richards as a language for wring operating systems and compilers. C was evolved from B by Dennis Ritchie at Bell Laboratories and it was implemented in 1972. It initially became widely known as the development language of the UNIX operating system. Lots of today's leading operating systems are written in C and C++. C language is mostly hardware independent as it is possible to write C programs that are portable to most computers.

Why we use c language C has been used successfully for each kind of programming problem thinkable from operating systems to spreadsheets to expert systems - and efficient compilers are accessible for machines ranging in power from the Apple Macintosh to the Cray supercomputers. The largest measure of C's success appears to be based on strictly sensible considerations:

1. The standard library concept.
2. The ease with that applications can be optimized by hand-

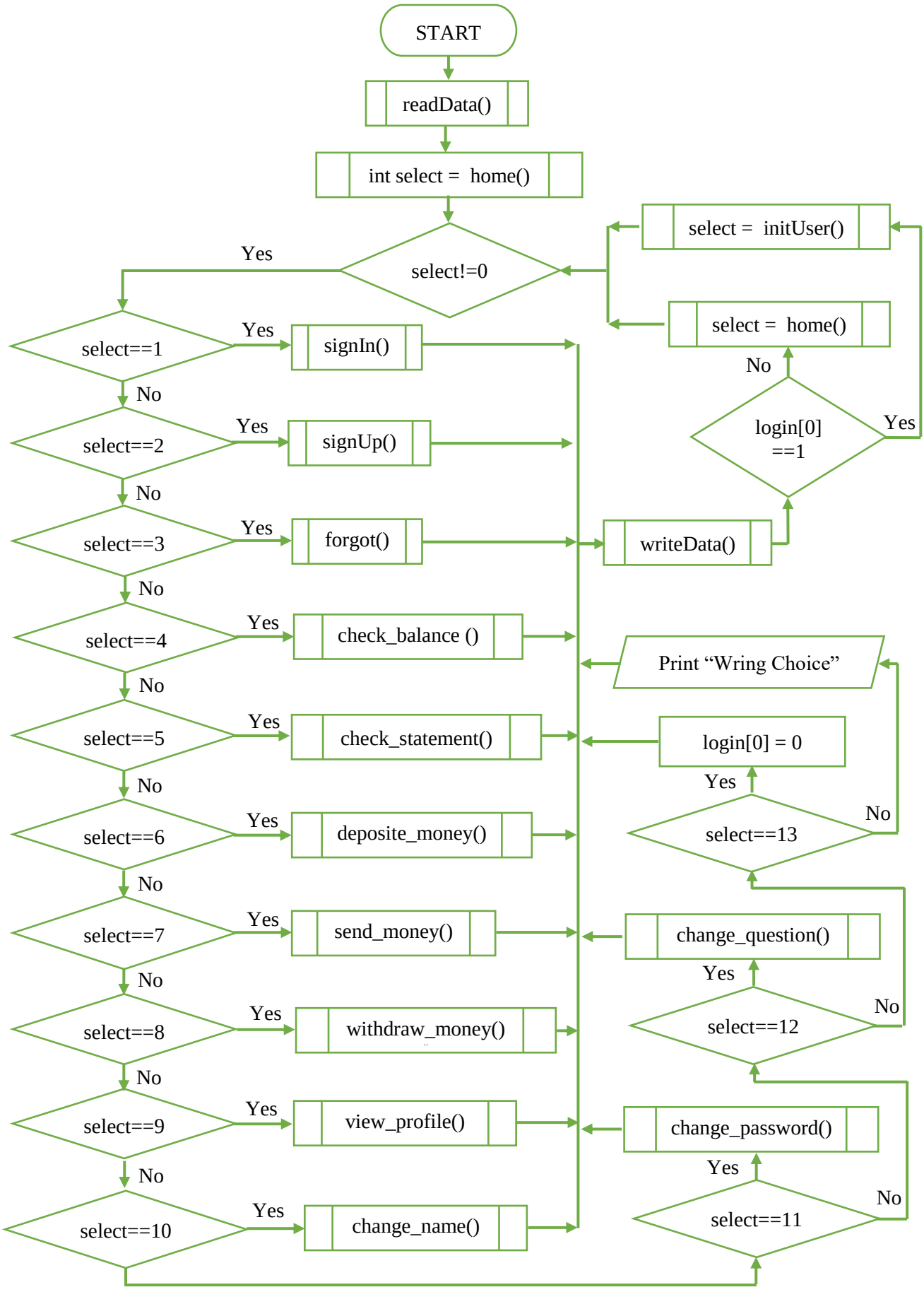
coding isolated procedures.
3. A powerful and varied repertoire of operators.
4. The portability of the compiler.

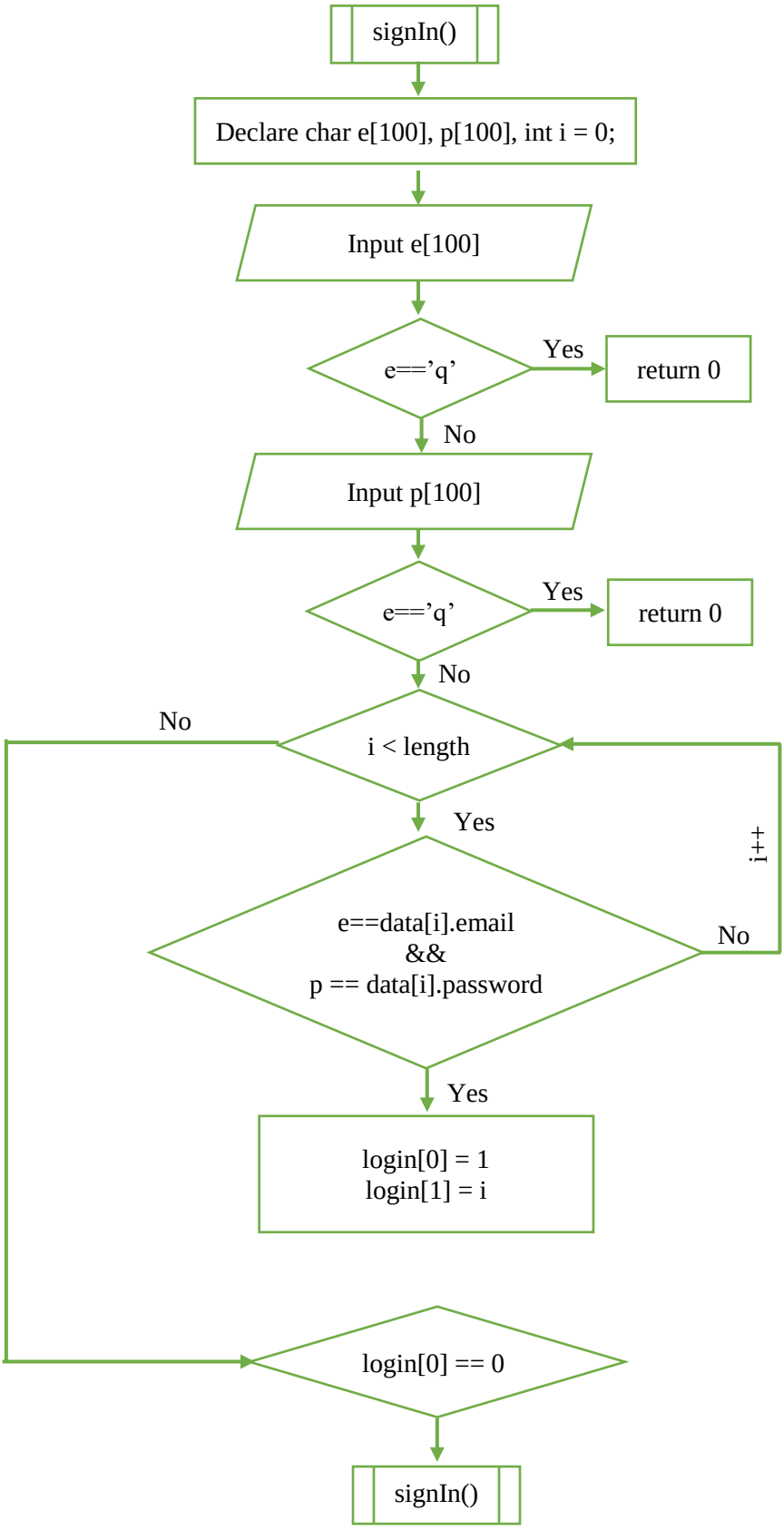
Design Goals/Objective

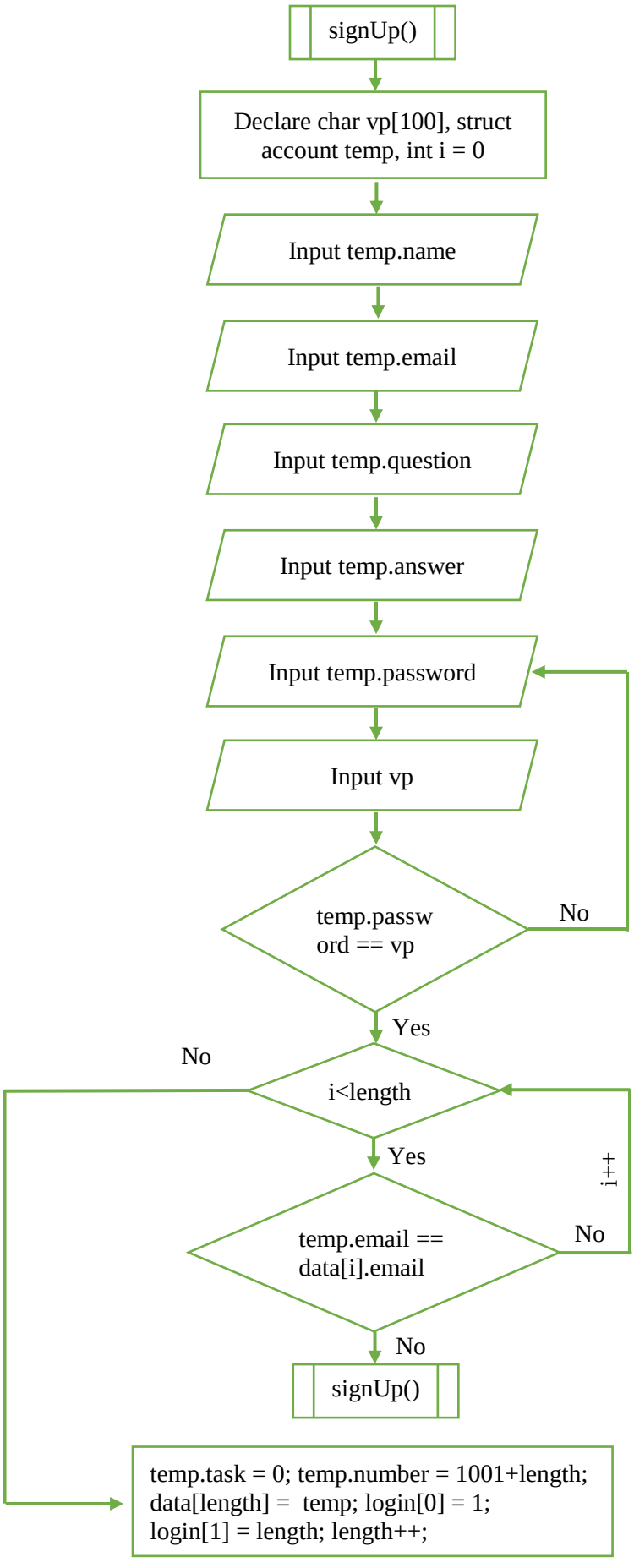
The goal of the project is to design a Bank Management System: Successful optimization of the “profitability-risk” ratio in bank lending operations is largely determined using effective bank management methods. The ability to take reasonable risks is one of the elements of entrepreneurship culture in general and banking culture.

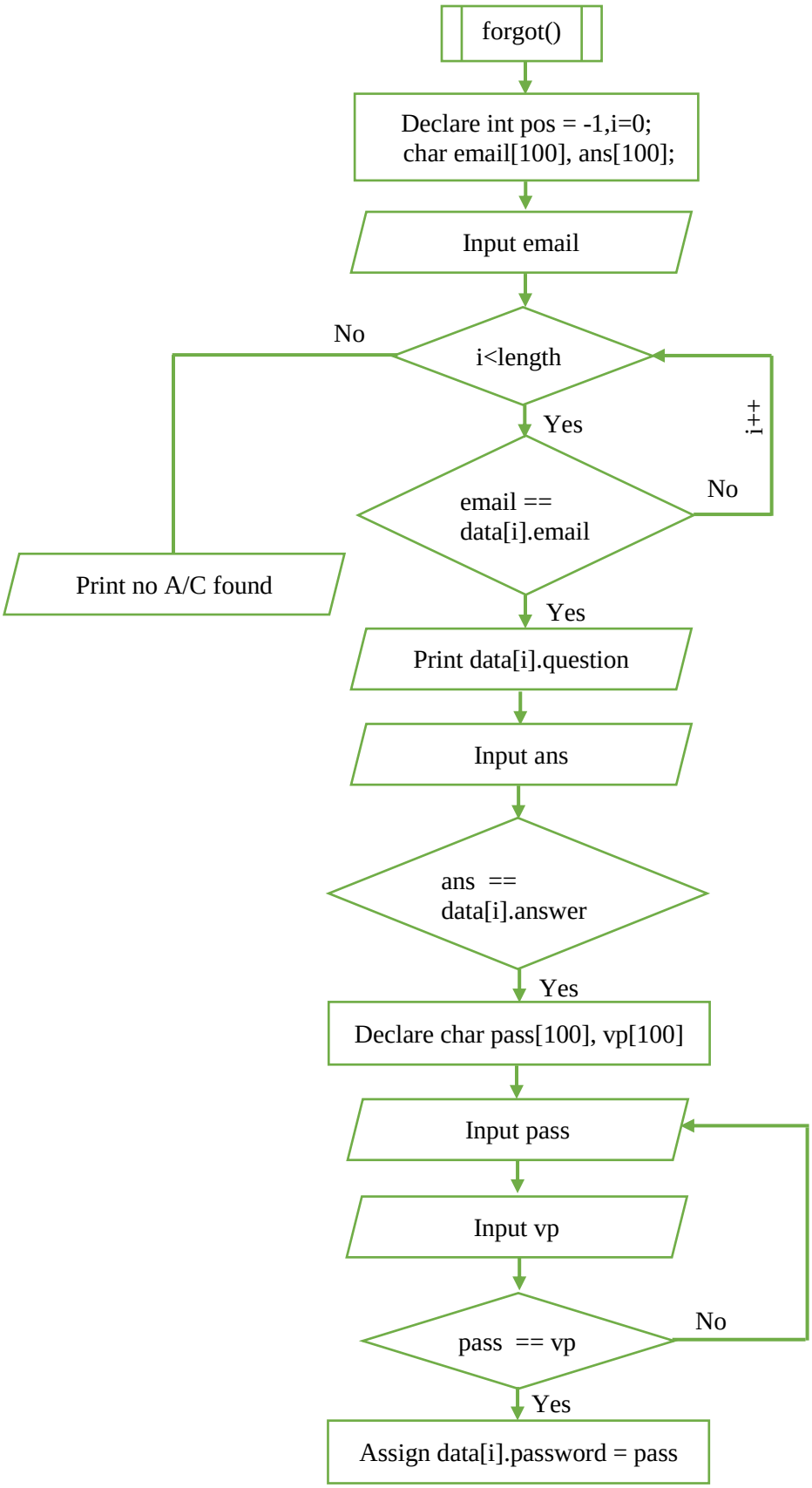
Flow Chart

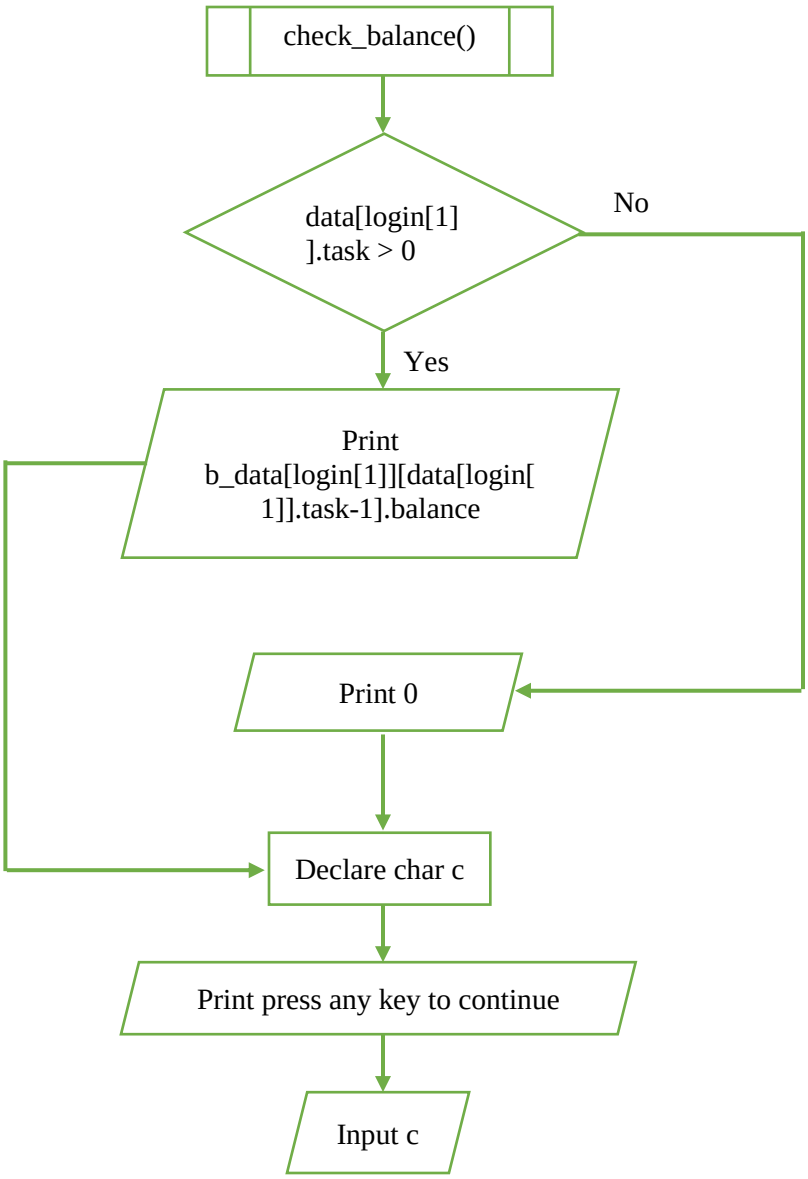
When the program is started, the user will direct to the main menu. The user will be required to select one of the four options if not logged in. If the user is logged in user required to select one of eleven options.

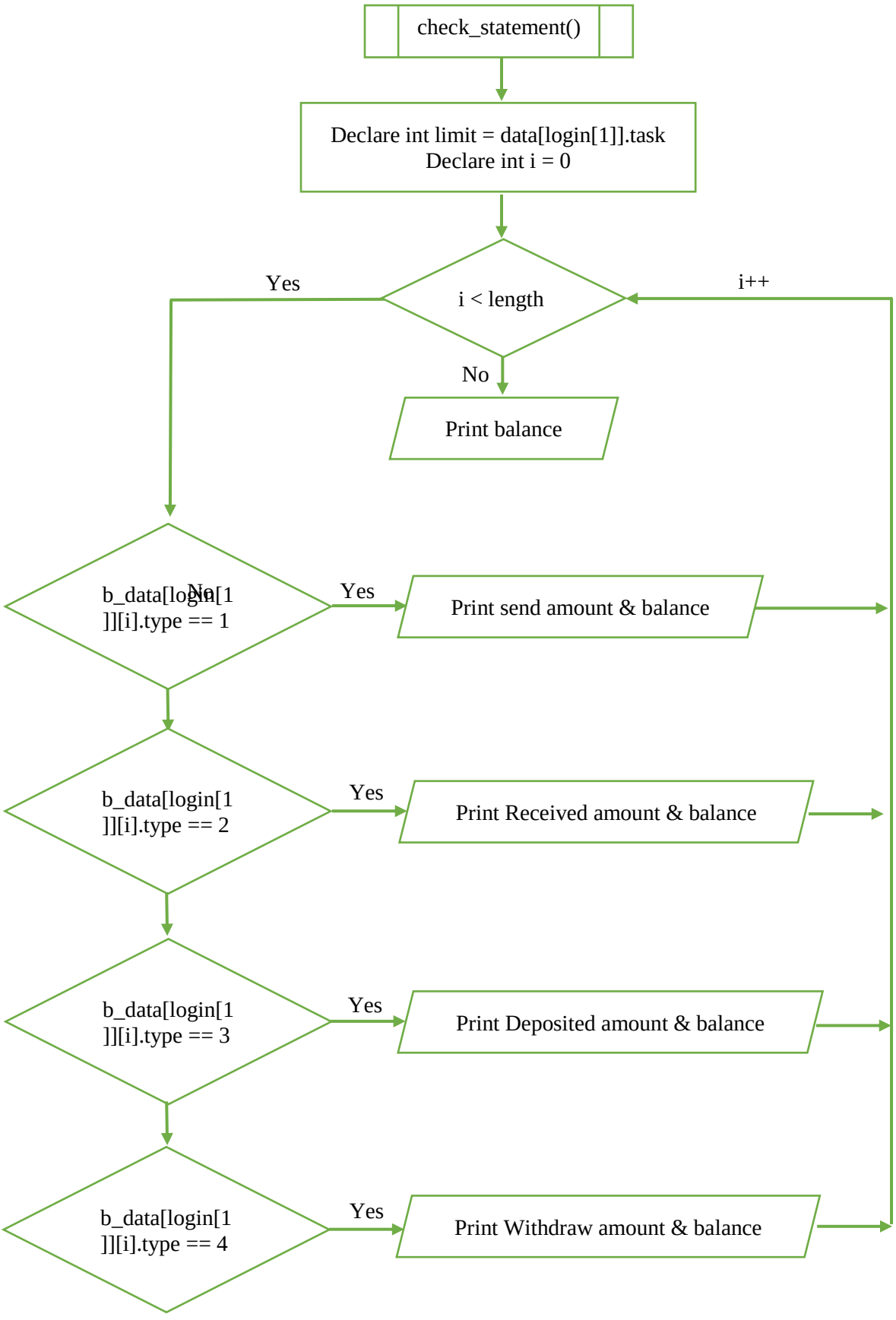

START
readData()
int select = home()
select!=0
Yes
select = initUser()
select = home()
No
select==1
Yes
signIn()
No
select==2
Yes
signUp()
No
select==3
Yes
forgot()
No
select==4
Yes
check_balance ()
No
select==5
Yes
check_statement()
No
select==6
Yes
deposite_money()
No
select==7
Yes
send_money()
No
select==8
Yes
withdraw_money()
No
select==9
Yes
view_profile()
No
select==10
Yes
change_name()
login[0] ==1
Yes
writeData()
Print “Wring Choice”
login[0] = 0
Yes
select==13
No
change_question()
Yes
select==12
No
change_password()
Yes
select==11
No

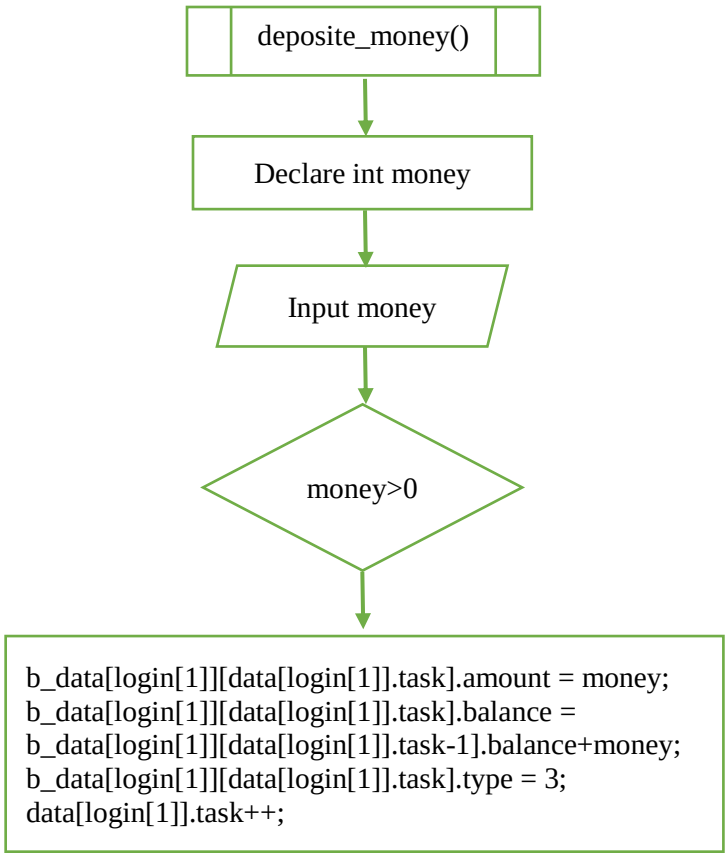

signIn()
Declare char e[100], p[100], int i = 0;
Input e[100]
e==’q’
Yes
return 0
No
Input p[100]
e==’q’
Yes
return 0
No
No
i < length
Yes
i++
e==data[i].email
&&
p == data[i].password
No
Yes
login[0] = 1
login[1] = i
login[0] == 0
signIn()

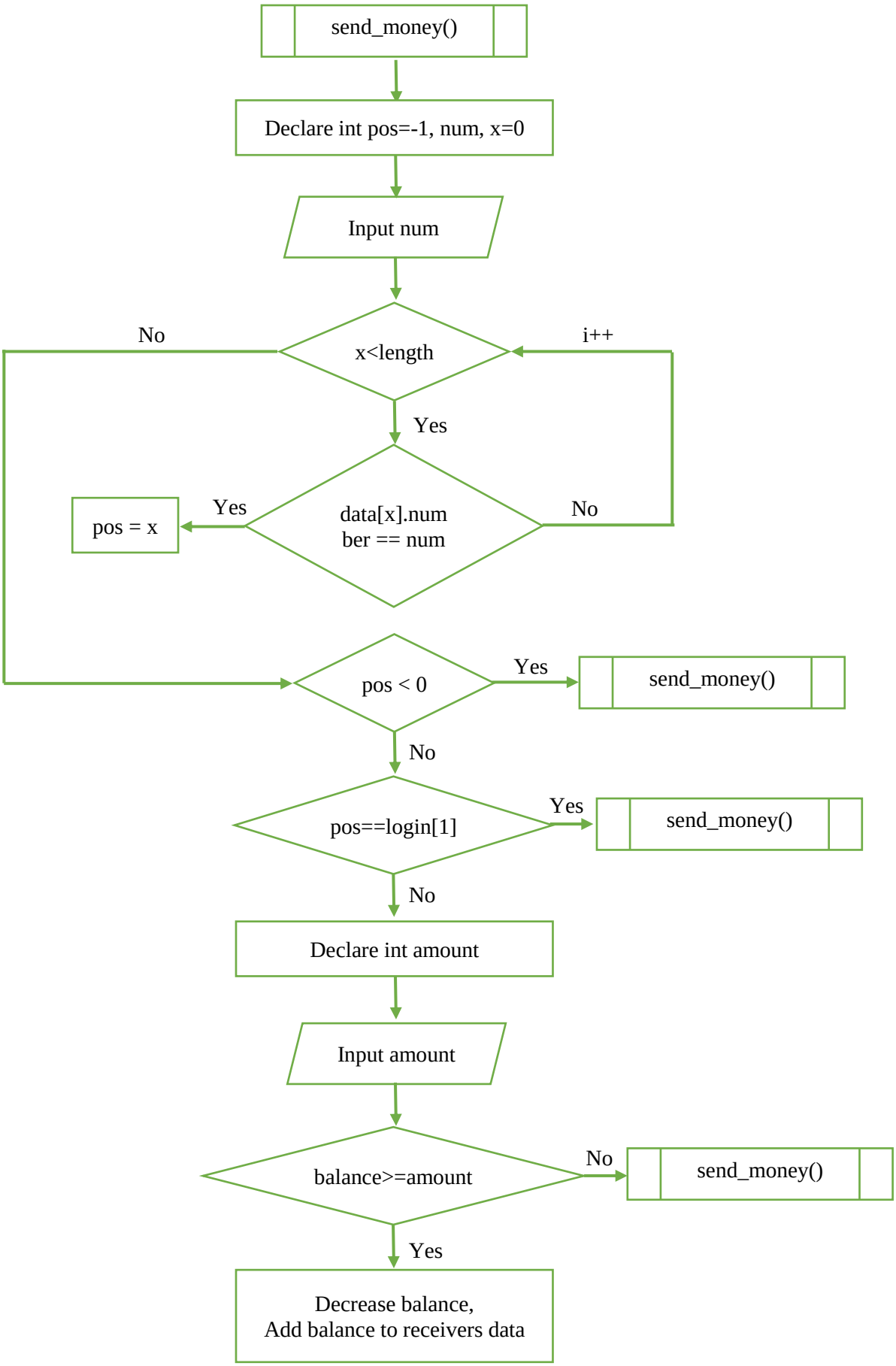

signUp()
Declare char vp[100], struct account temp, int i = 0
Input temp.name
Input temp.email
Input temp.question
Input temp.answer
Input temp.password
Input vp
temp.passw ord == vp
No
Yes
No
i<length
Yes
i++
temp.email == data[i].email
No
No
signUp()
temp.task = 0; temp.number = 1001+length; data[length] = temp; login[0] = 1; login[1] = length; length++;

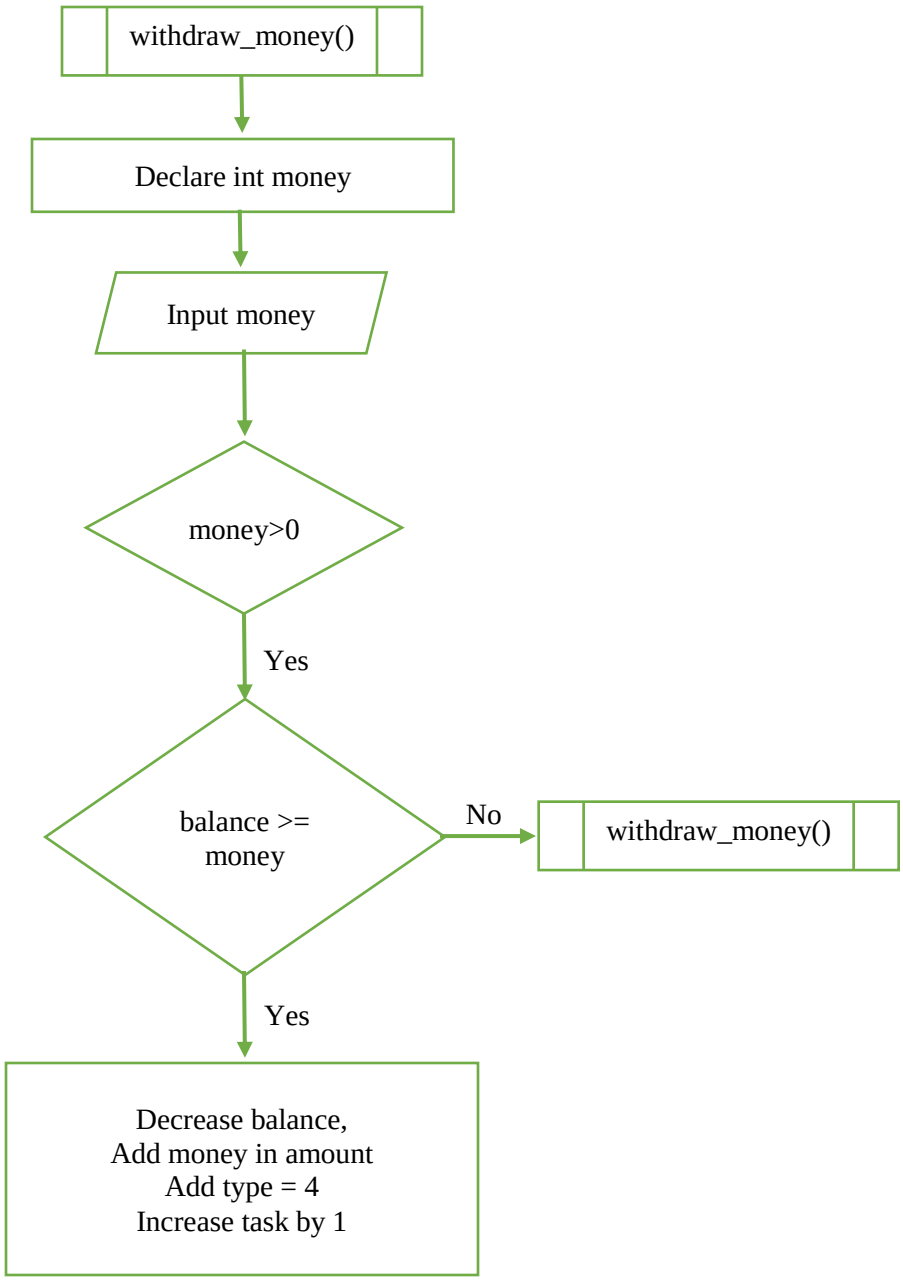

forgot()
Declare int pos = -1,i=0;
char email[100], ans[100];
Input email
i<length
No
Yes
i++
email ==
data[i].email
No
Print no A/C found
Yes
Print data[i].question
Input ans
ans ==
data[i].answer
Yes
Declare char pass[100], vp[100]
Input pass
Input vp
pass == vp
No
Yes
Assign data[i].password = pass

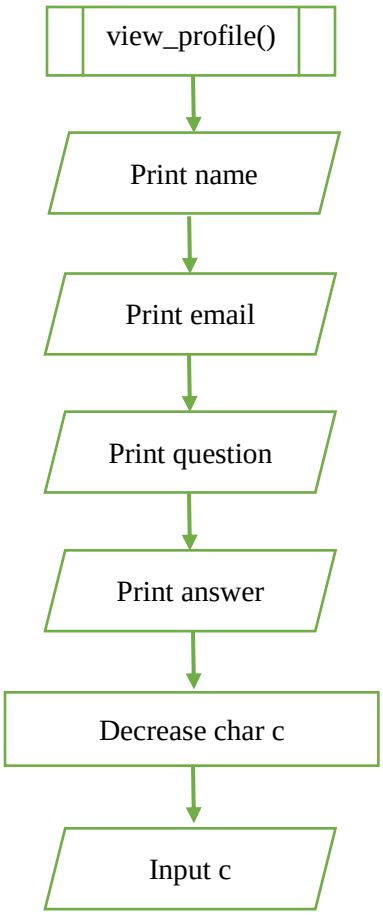

check_balance()
data[login[1]
].task > 0
No
Yes
Print
b_data[login[1]][data[login[
1]].task-1].balance
Print 0
Declare char c
Print press any key to continue
Input c

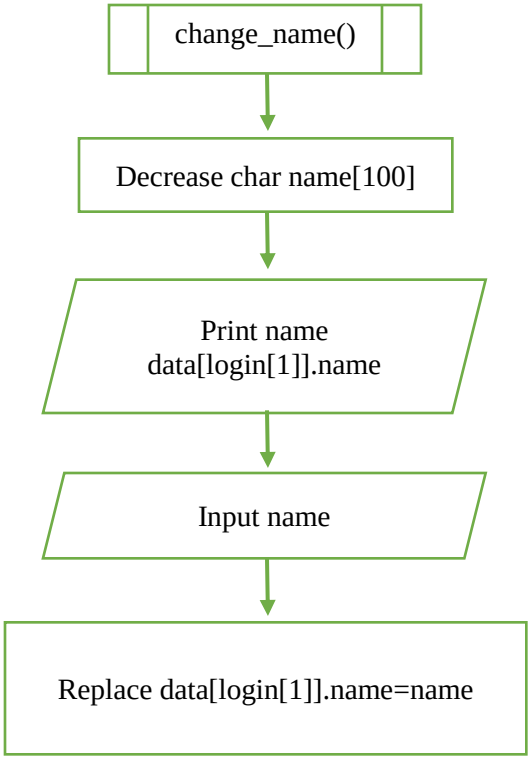

check_statement()
Declare int limit = data[login[1]].task
Declare int i = 0
i < length
Yes
No
i++
Print balance
b_data[login[1
]][i].type == 1
No
Yes
Print send amount & balance
b_data[login[1
]][i].type == 2
Yes
Print Received amount & balance
b_data[login[1
]][i].type == 3
Yes
Print Deposited amount & balance
b_data[login[1
]][i].type == 4
Yes
Print Withdraw amount & balance

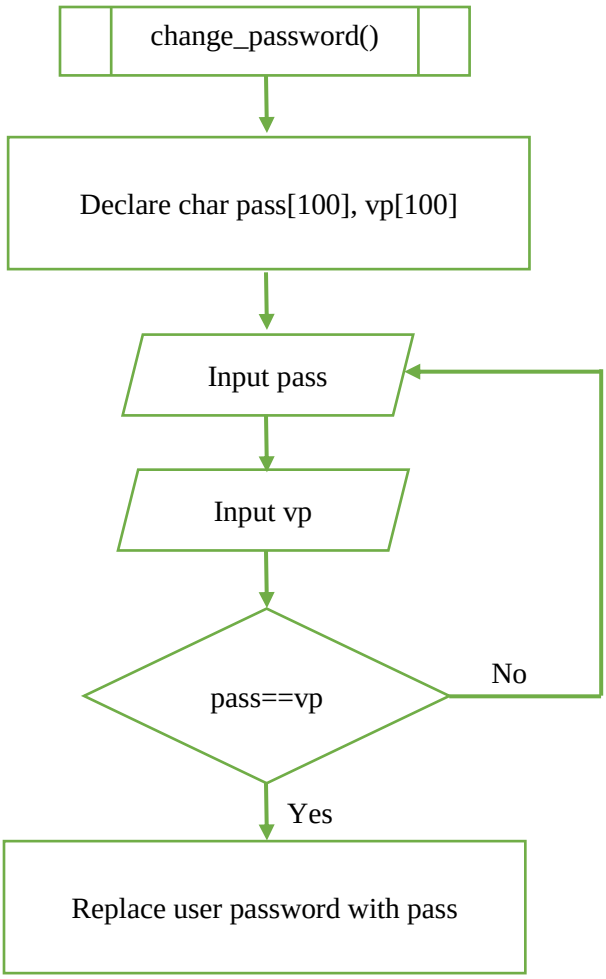

deposite_money()
Declare int money
Input money
money>0
b_data[login[1]][data[login[1]].task].amount = money;
b_data[login[1]][data[login[1]].task].balance =
b_data[login[1]][data[login[1]].task-1].balance+money;
b_data[login[1]][data[login[1]].task].type = 3;
data[login[1]].task++;

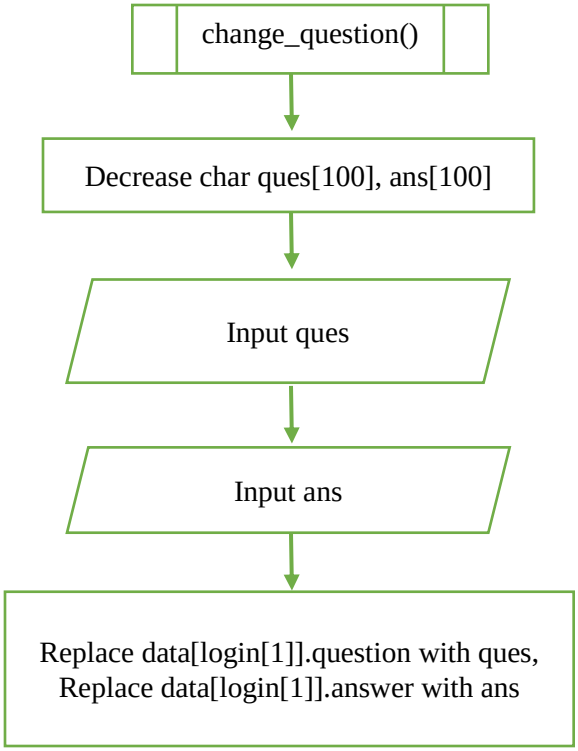

send_money()
Declare int pos=-1, num, x=0
Input num
x<length
No
i++
Yes
data[x].num
ber == num
Yes
pos = x
No
pos < 0
Yes
send_money()
No
pos==login[1]
Yes
send_money()
No
Declare int amount
Input amount
balance>=amount
No
send_money()
Yes
Decrease balance,
Add balance to receivers data


withdraw_money()
Declare int money
Input money
money>0
Yes
balance >= money
No
withdraw_money()
Yes
Decrease balance,
Add money in amount
Add type = 4
Increase task by 1


view_profile()
Print name
Print email
Print question
Print answer
Decrease char c
Input c


change_name()
Decrease char name[100]
Print name
data[login[1]].name
Input name
Replace data[login[1]].name=name


change_password()
Declare char pass[100], vp[100]
Input pass
Input vp
pass==vp
No
Yes
Replace user password with pass


change_question()
Decrease char ques[100], ans[100]
Input ques
Input ans
Replace data[login[1]].question with ques,
Replace data[login[1]].answer with ans

Implementation

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

#define _100 100
#define _1000 1000

int length = 0;
int login[2] = {0,0};

struct account {
    int task;
    int number;
    char name[_100];
    char email[_100];
    char answer[_1000];
    char password[_100];
    char question[_1000];
};

/*
* type 1 = send
* type 2 = received
* type 3 = deposite
* type 4 = withdraw
*/
struct balance {
    int balance;
    int amount;
    int type;
    char date[_100];
};

struct account data[_100] = {0};
```

```
struct balance b_data[_100][_100] = {0};

int home() {
   int c = 0;
   printf("**********************************************************
*****************\n");
   printf("*\t\t\tWelcome to Bangladesh Bank\t\t\t*\n");
   printf("**********************************************************
*****************\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\t1. Sign in to your account\t\t\t\t\t*\n");
   printf("*\t2. Create a new account\t\t\t\t\t*\n");
   printf("*\t3. Forgot password\t\t\t\t\t*\n");
   printf("*\t4. Exit the Application\t\t\t\t\t*\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\tEnter your choice: ");
   fflush(stdin);
   scanf("%d", &c);
   if (c<1 || c>4) c=-1;
   if(c==4) c=0;
   return c;
}

int initUser() {
   int c = 0;
   printf("**********************************************************
*****************\n");
   printf("*\t\t\tWelcome to Bangladesh Bank\t\t\t*\n");
   printf("**********************************************************
*****************\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\tHi, %s\n*\tA/C: %d\t\t\t\t\n", data[login[1]].name,
data[login[1]].number);
   printf("*\t\t\t\t\t\t\t\t\t*\n*");
   printf("\033[0;32m");
   printf("\tManage Funds\t\t\t\t\t\t");
   printf("\033[0m");
   printf("*\n*");
   printf("\033[0;32m");
   printf("\t----------\t\t\t\t\t\t");
```

```
    printf("\033[0m");
    printf("*\n");
    printf("*\t1. Check your account Balance\t\t\t\t*\n");
    printf("*\t2. Check your account Statement\t\t\t\t*\n");
    printf("*\t3. Deposite money\t\t\t\t\t*\n");
    printf("*\t4. Send money\t\t\t\t\t\t*\n");
    printf("*\t5. Withdraw money\t\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n*");
    printf("\033[0;32m");
    printf("\tManage Account\t\t\t\t\t\t");
    printf("\033[0m");
    printf("*\n*");
    printf("\033[0;32m");
    printf("\t--------------\t\t\t\t\t\t");
    printf("\033[0m");
    printf("*\n");
    printf("*\t6. View your profile\t\t\t\t\t*\n");
    printf("*\t7. Change your account name\t\t\t\t*\n");
    printf("*\t8. Change your password\t\t\t\t\t*\n");
    printf("*\t9. Change your Security Question\t\t\t*\n");
    printf("*\t10. Logout\t\t\t\t\t\t*\n");
    printf("*\t11. Exit the Application\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter your choice: ");
    fflush(stdin);
    scanf("%d", &c);
    if (c<1 || c>11) c=-1;
    if(c==11) c=0;
    else c += 3;
    return c;
}

int signIn() {
    login[0] = 0;
    char e[100], p[100];
    printf("**********************************************************
**********\n");
    printf("*\t\t\t    Enter (q) to exit    \t\t\t*\n");
    printf("**********************************************************
**********\n");
```

```
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter your Email: ");
    fflush(stdin);
    gets(e);
    if (strcmp(e, "q")==0)
        return 0;
    printf("*\tEnter your Password: ");
    gets(p);
    if (strcmp(p, "q")==0)
        return 0;
    for (int i = 0; i < 100; i++) {
        if (strcmp(e, data[i].email)==0 && strcmp(p, data[i].password)==0) {
            login[0] = 1;
            login[1] = i;
            break;
        }
    }
    if (login[0] == 0) {
        system("cls");
        printf("**********************************************************
**********************\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\tWrong Email or Password\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        signIn();
    }
    return 0;
}

int signUp() {
    char vp[100];
    struct account temp;
    printf("**********************************************************
**********************\n");
    printf("*\t\t\t    Create a new account     \t\t\t*\n");
    printf("**********************************************************
**********************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
```

```
printf("*\tEnter your Name: ");
fflush(stdin);
gets(temp.name);
if (strcmp(temp.name, "q")==0)
   return 0;
printf("*\tEnter your Email: ");
gets(temp.email);
if (strcmp(temp.email, "q")==0)
   return 0;
printf("*\tEnter your Security Question: ");
gets(temp.question);
if (strcmp(temp.question, "q")==0)
   return 0;
printf("*\tEnter your Answer: ");
gets(temp.answer);
if (strcmp(temp.answer, "q")==0)
   return 0;
printf("*\tEnter your password: ");
gets(temp.password);
if (strcmp(temp.password, "q")==0)
   return 0;
printf("*\tEnter your password again: ");
gets(vp);
if (strcmp(vp, "q")==0)
   return 0;
while (strcmp(temp.password, vp)) {
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\t");
   printf("\033[0;31m");
   printf("Password mismatch! Try again.");
   printf("\033[0m");
   printf("\t\t\t\t\t*\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\tEnter your password: ");
   gets(temp.password);
   if (strcmp(temp.password, "q")==0)
   return 0;
   printf("*\tEnter your password again: ");
   gets(vp);
   if (strcmp(vp, "q")==0)
```

```
        return 0;
    }

    for (int i = 0; i < length; i++) {
        if (strcmp(temp.email, data[i].email)==0) {
            char c;
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("\n*\tEmail already exist\t\t\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("\n*\tWant to try again? (y or n):");
            scanf("%c", &c);
            if (c=='y' || c=='Y') {
                system("cls");
                signUp();
            }
            return 0;
        }
    }
    temp.task = 0;
    temp.number = 1001+length;
    data[length] =  temp;
    login[0] = 1;
    login[1] = length;
    length++;
    return 0;
}

int forgot() {
    int pos = -1;
    char email[100], ans[100];
    printf("**************************************************************
******************\n");
    printf("*\t\t\t    Recover your account     \t\t\t*\n");
    printf("**************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter your Email: ");
    fflush(stdin);
```

```
gets(email);
if (strcmp(email, "q")==0)
   return 0;
for (int i = 0; i < length; i++) {
   if (strcmp(email, data[i].email)==0) {
      printf("*\t %s: ", data[i].question);
      fflush(stdin);
      gets(ans);
      if (strcmp(ans, data[i].answer)==0) {
         char pass[_100], vp[_100];
         printf("*\tEnter your password: ");
         gets(pass);
         if (strcmp(pass, "q")==0)
            return 0;
         printf("*\tEnter your password again: ");
         gets(vp);
         if (strcmp(vp, "q")==0)
            return 0;
         while (strcmp(pass, vp)) {
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\t");
            printf("\033[0;31m");
            printf("Password mismatch! Try again.");
            printf("\033[0m");
            printf("\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\tEnter your password: ");
            gets(pass);
            if (strcmp(pass, "q")==0)
            return 0;
            printf("*\tEnter your password again: ");
            gets(vp);
            if (strcmp(vp, "q")==0)
            return 0;
         }
         strncpy(data[i].password, pass, _100);
      }
      return 0;
   }
}
```

```
    char c;
    printf("*\tNo Account found with %s\n", email);
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("\n*\tWant to try again? (y or n):");
    scanf("%c", &c);
    if (c=='y' || c=='Y') {
        system("cls");
        forgot();
    }
}

void check_balance() {
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\tEnter (q) to exit\t\t\t*\n");
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tYour Current balance is %d\n", data[login[1]].task > 0 ?
b_data[login[1]][data[login[1]].task-1].balance : 0);
    printf("*\t\t\t\t\t\t\t\t\t*\n\n");
    printf("*\tPress any key to continue...");
    char c;
    fflush(stdin);
    scanf("%c", &c);
}

void check_statement() {
    printf("**********************************************************************
******************\n");
    printf("*\t\t\tSatement Generated   \t\t\t*\n");
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n*");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*");
    printf("\033[0;31m");
    printf("\t-----------------------------------------------\t\t");
    printf("\033[0m");
```

```
    printf("*\n*");
    printf("\033[0;31m");
    printf("\t|\ttype\t|\tamount\t|\tbalance\t|\t\t");
    printf("\033[0m");
    printf("*\n*");
    printf("\033[0;31m");
    printf("\t-----------------------------------------------\t\t");
    printf("\033[0m");
    printf("*\n");
    int limit = data[login[1]].task;
    for (int i = 0; i < limit; i++) {
        switch (b_data[login[1]][i].type) {
            case 1:
                printf("*\t|\tSend\t|\t%d\t|\t%d\t|\t\t*", b_data[login[1]][i].amount,
b_data[login[1]][i].balance);
                break;
            case 2:
                printf("*\t|    Received   |\t%d\t|\t%d\t|\t\t*",
b_data[login[1]][i].amount, b_data[login[1]][i].balance);
                break;
            case 3:
                printf("*\t|   Deposited   |\t%d\t|\t%d\t|\t\t*",
b_data[login[1]][i].amount, b_data[login[1]][i].balance);
                break;
            case 4:
                printf("*\t|    Withdraw   |\t%d\t|\t%d\t|\t\t*",
b_data[login[1]][i].amount, b_data[login[1]][i].balance);
                break;
            default:
                break;
        };
        printf("\n*\t-----------------------------------------------\t\t*\n");
    }
    printf("*\t\t\t\t\t\t\t\t\t*\n*");
    printf("\033[0;32m");
    printf("\tYour Current balance is: %d\n", data[login[1]].task > 0 ?
b_data[login[1]][data[login[1]].task-1].balance : 0);
    printf("\033[0m");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tPress any key to continue...");
```

```
    char c;
    fflush(stdin);
    scanf("%c", &c);
}

void deposite_money() {
    int money;
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\tEnter (0) to exit\t\t\t*\n");
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter an amount to deposite: ");
    scanf("%d", &money);
    if (money>0) {
        b_data[login[1]][data[login[1]].task].amount = money;
        b_data[login[1]][data[login[1]].task].balance =
b_data[login[1]][data[login[1]].task-1].balance+money;
        b_data[login[1]][data[login[1]].task].type = 3;
        data[login[1]].task++;
    }
}

void send_money() {
    int pos=-1, num;
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t    Send Money     \t\t\t*\n");
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter acccount number: ");
    fflush(stdin);
    scanf("%d", &num);
    for (int x=0; x<length; x++) {
        if (data[x].number == num) {
            pos = x;
```

```
            break;
        }
    }
    if (pos<0) {
        char c;
        system("cls");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t    Send Money     \t\t\t*\n");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\tNo Account found with %d\n", num);
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("\n*\tWant to try again? (y or n):");
        fflush(stdin);
        scanf("%c", &c);
        if (c=='y' || c=='Y') {
            system("cls");
            send_money();
        }
    } else if (pos==login[1]) {
        char c;
        system("cls");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t    Send Money     \t\t\t*\n");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\tCannot send money to your own account   \t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("\n*\tWant to try again? (y or n):");
        fflush(stdin);
        scanf("%c", &c);
```

```
        if (c=='y' || c=='Y') {
            system("cls");
            send_money();
        }
    } else {
        int amount;
        struct balance temp;
        system("cls");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t    Send Money     \t\t\t*\n");
        printf("**********************************************************
********************\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\tAccount holder name: %s", data[pos].name);
        printf("\n*\tEnter amount: ");
        fflush(stdin);
        scanf("%d", &amount);
        if (data[login[1]].task>0 && b_data[login[1]][data[login[1]].task-
1].balance>=amount) {
            if (data[pos].task > 0) {
              b_data[pos][data[pos].task].balance = b_data[pos][data[pos].task-
1].balance+amount;
            } else {
               b_data[pos][data[pos].task].balance = amount;
            }
            b_data[pos][data[pos].task].amount = amount;
            b_data[login[1]][data[login[1]].task].balance =
b_data[login[1]][data[login[1]].task-1].balance-amount;
            b_data[login[1]][data[login[1]].task].amount = amount;
            b_data[login[1]][data[login[1]].task].type = 1;
            b_data[pos][data[pos].task].type = 2;
            data[login[1]].task++;
            data[pos].task++;
        } else {
            char c;
            system("cls");
```

```
            printf("**********************************************************
**********************\n");
            printf("*\t\t\t    Send Money     \t\t\t*\n");
            printf("**********************************************************
**********************\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("\n*\t\t   Insufficient funds.\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\tWant to try again? (y or n): ");
            fflush(stdin);
            scanf("%c", &c);
            if (c=='y' || c=='Y') {
               system("cls");
               send_money();
            }
         }
      }
   }

void withdraw_money() {
   int money;
   printf("**********************************************************
**********************\n");
   printf("*\t\t\t    Withdraw Money     \t\t\t*\n");
   printf("**********************************************************
**********************\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("*\tEnter an amount to withdraw: ");
   scanf("%d", &money);
   if (money>0) {
      if (data[login[1]].task > 0 && b_data[login[1]][data[login[1]].task-
1].balance >= money) {
         b_data[login[1]][data[login[1]].task].balance =
b_data[login[1]][data[login[1]].task-1].balance-money;
```

```
            b_data[login[1]][data[login[1]].task].amount = money;
            b_data[login[1]][data[login[1]].task].type = 4;
            data[login[1]].task++;
        } else {
            char c;
            system("cls");
            printf("**********************************************************
**********************\n");
            printf("*\t\t\t    Withdraw Money     \t\t\t*\n");
            printf("**********************************************************
**********************\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("\n*\t\t   Insufficient funds.\t\t\t\t*\n");
            printf("*\t\t\t\t\t\t\t\t\t*\n");
            printf("*\tWant to try again? (y or n): ");
            fflush(stdin);
            scanf("%c", &c);
            if (c=='y' || c=='Y') {
                system("cls");
                withdraw_money();
            }
        }

    }
}

void view_profile() {
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\tProfile\t\t\t\t*\n");
    printf("**********************************************************************
******************\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tName: %s\n", data[login[1]].name);
    printf("*\tEmail: %s\n", data[login[1]].email);
    printf("*\tQuestion: %s\n", data[login[1]].question);
    printf("*\tAnswer: %s\n", data[login[1]].answer);
```

```
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tPress any key to continue...");
    char c;
    fflush(stdin);
    scanf("%c", &c);
}

void change_name() {
    char name[_100];
    printf("**********************************************************
**********\n");
    printf("*\t\t\t    Change Name     \t\t\t*\n");
    printf("**********************************************************
**********\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tYour name is: %s\n", data[login[1]].name);
    printf("*\tEnter your currect name: ");
    fflush(stdin);
    gets(name);
    strncpy(data[login[1]].name, name, _100);
}

int change_password() {
    char pass[_100], vp[_100];
    printf("**********************************************************
**********\n");
    printf("*\t\t\t    Change Name     \t\t\t*\n");
    printf("**********************************************************
**********\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\tEnter your password: ");
    fflush(stdin);
    gets(pass);
```

```
    if (strcmp(pass, "q")==0)
        return 0;
    printf("*\tEnter your password again: ");
    fflush(stdin);
    gets(vp);
    if (strcmp(vp, "q")==0)
        return 0;
    while (strcmp(pass, vp)) {
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\t");
        printf("\033[0;31m");
        printf("Password mismatch! Try again.");
        printf("\033[0m");
        printf("\t\t\t\t\t*\n");
        printf("*\t\t\t\t\t\t\t\t\t*\n");
        printf("*\tEnter your password: ");
        fflush(stdin);
        gets(pass);
        if (strcmp(pass, "q")==0)
            return 0;
        printf("*\tEnter your password again: ");
        fflush(stdin);
        gets(vp);
        if (strcmp(vp, "q")==0)
            return 0;
    }
    strncpy(data[login[1]].password, pass, _100);
}

void change_question() {
    char ques[_100], ans[_100];
    printf("**********************************************************
**********\n");
    printf("*\t\t\tChange Security Question\t\t\t*\n");
    printf("**********************************************************
**********\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\tEnter (q) to exit\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
    printf("*\t\t\t\t\t\t\t\t\t*\n");
```

```
    printf("*\tEnter your question: ");
    fflush(stdin);
    gets(ques);
    printf("*\tEnter your answer: ");
    fflush(stdin);
    gets(ans);
    strncpy(data[login[1]].question, ques, _1000);
    strncpy(data[login[1]].answer, ans, _1000);
}

int readData() {
    FILE* account_file = fopen("account.txt", "r");
    int i = 0;
    while (fscanf(account_file, "%d %d %s %s %s %s %s", &data[i].task,
&data[i].number,
     &data[i].name, &data[i].email, &data[i].answer, &data[i].password,
&data[i].question)) {
        if (data[i].number==0)
            return 0;
        i++;
    }
    length = i+1;
}

char* int_to_string(int num) {
    char temp[10] = {""};
    static char str[100] = {""};
    while (num) {
        temp[0] = (num%10)+'0';
        strcat(str, temp);
        num=num/10;
    }
    strrev(str);
    return str;
}

void saveData() {
    int i, j;
    FILE *account_file = fopen("account.txt", "w");
    FILE *balance_file = fopen("balance_file.txt", "w");
```

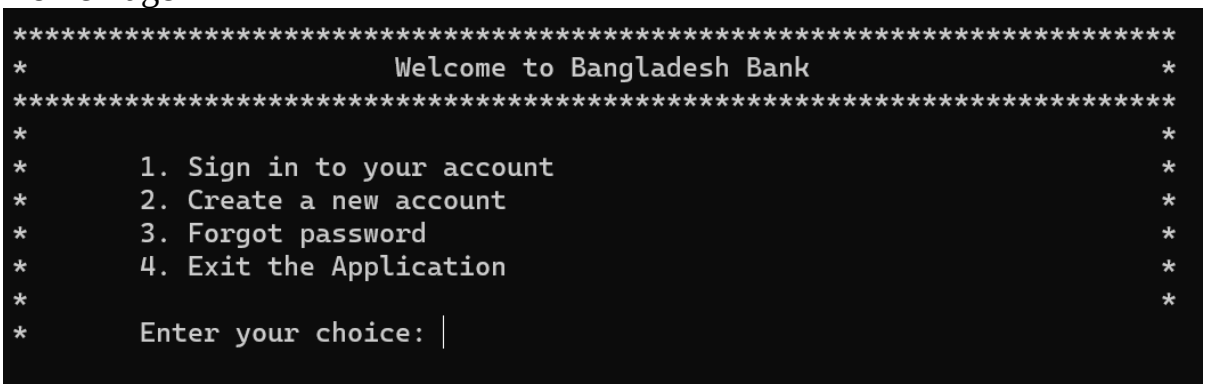
```
    for (i = 0; i < _100; i++) {
        fprintf(account_file, "%d %d %s %s %s %s %s\n", data[i].task,
data[i].number,
         data[i].name, data[i].email, data[i].answer, data[i].password,
data[i].question);
        for(j = 0; j < data[i].task; i++) {
            fprintf(balance_file, "%d %d %d %s", b_data[i][j].balance,
                b_data[i][j].amount, b_data[i][j].type, b_data[i][j].date);
        }
    }
    fclose(account_file);
    char a;
    getc(a);
}

int main() {
    readData();
    int select = home();
    while (select) {
        system("cls");
        switch (select) {
            case 1:
                signIn();
                break;
            case 2:
                signUp();
                break;
            case 3:
                forgot();
                break;
            case 4:
                check_balance();
                break;
            case 5:
                check_statement();
                break;
            case 6:
                deposite_money();
                break;
            case 7:
```

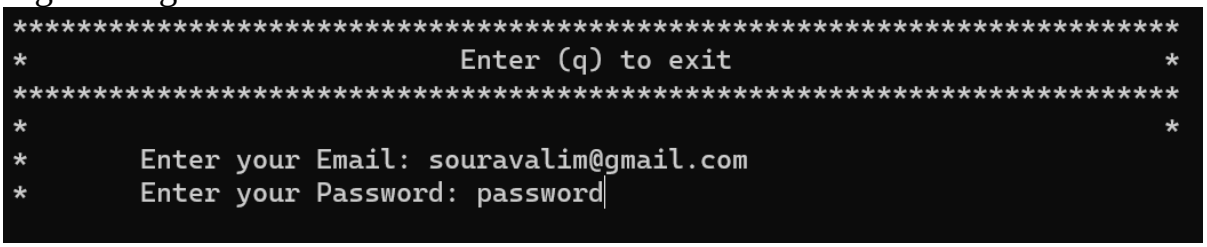
```
            send_money();
            break;
        case 8:
            withdraw_money();
            break;
        case 9:
            view_profile();
            break;
        case 10:
            change_name();
            break;
        case 11:
            change_password();
            break;
        case 12:
            change_question();
            break;
        case 13:
            login[0] = 0;
            break;
        default:
            printf("*\t\t\tWrong choice\t\t\t*\n");
            break;
      }
      system("cls");
      if (login[0]==1)
         select = initUser();
      else select = home();
      saveData();
   }
   printf("*\t\t\t\t\t\t\t\t\t*\n");
   printf("**********************************************************
*****************\n");
   printf("*\t\tThank you for using our application\t\t\t*\n");
   printf("**********************************************************
*****************\n\n");
   char a;
   scanf("%c", &a);
   return 0;
}
```

Screenshots

1. Home Page

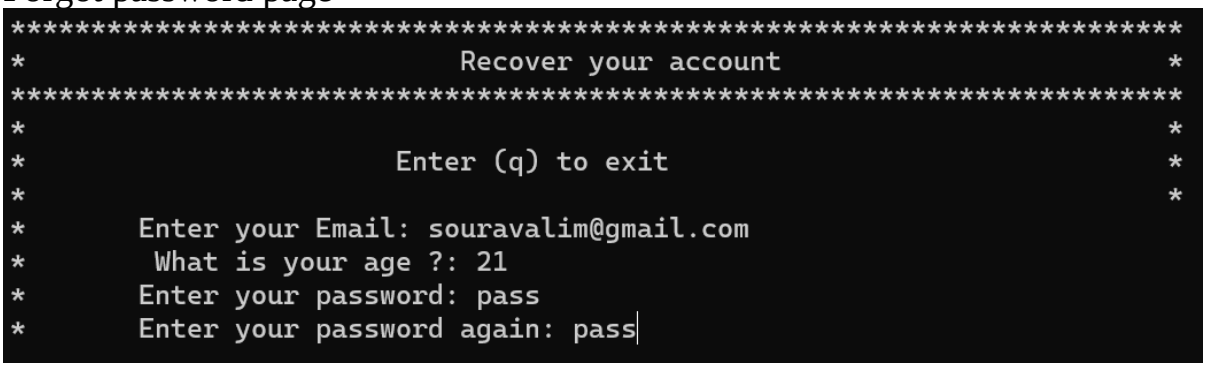


2. Sign In Page

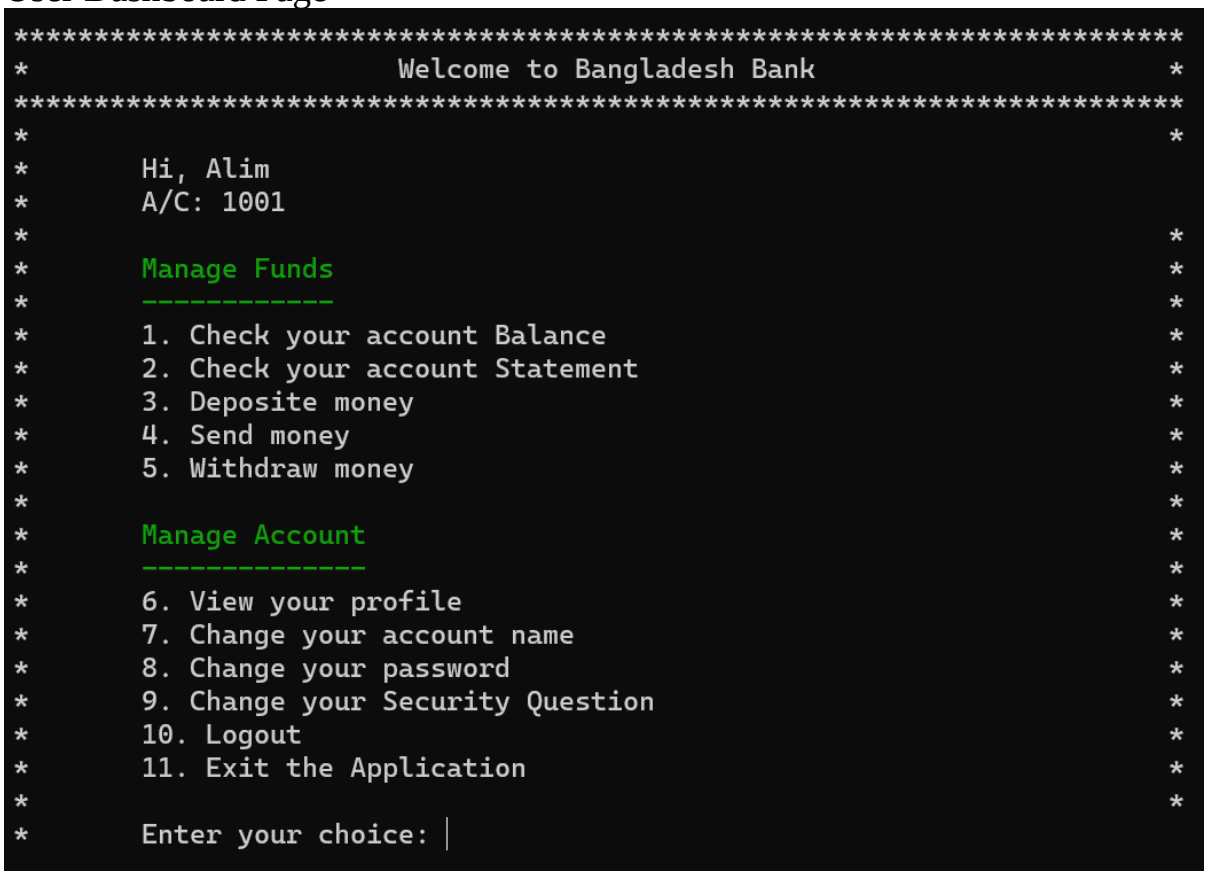


3. Create account page

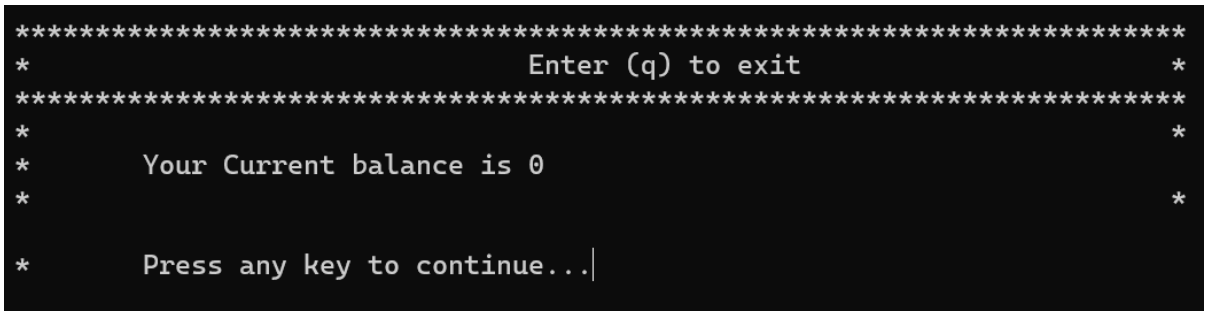
4. Forgot password page



5. User Dashboard Page



6. Check account balance



7. Check Statement

```
**************************************************************************
*                      Satement Generated                                *
**************************************************************************
*                                                                        *
**                                                                       *
*       ------------------------------------------------                 *
*       |       type    |       amount  |       balance |                *
*       ------------------------------------------------                 *
*       |   Deposited   |       455     |       455     |                *
*       ------------------------------------------------                 *
*       |       Send    |       67      |       388     |                *
*       ------------------------------------------------                 *
*       |    Withdraw   |       77      |       311     |                *
*       ------------------------------------------------                 *
*                                                                        *
*       Your Current balance is: 311                                     *
*                                                                        *
*       Press any key to continue...
```

8. Deposite money

```
**************************************************************************
*                              Enter (0) to exit                         *
**************************************************************************
*                                                                        *
*       Enter an amount to deposite: 100
```

9. Send money

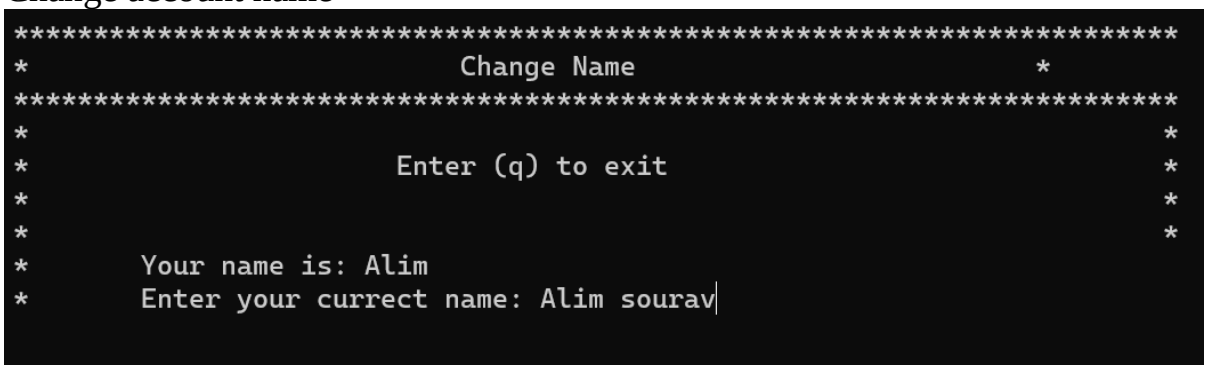
```
**************************************************************************
*                            Send Money                                  *
**************************************************************************
*                                                                        *
*                       Enter (q) to exit                                *
*                                                                        *
*                                                                        *
*       Account holder name: Alim
*       Enter amount: 100
```

10. Withdraw money

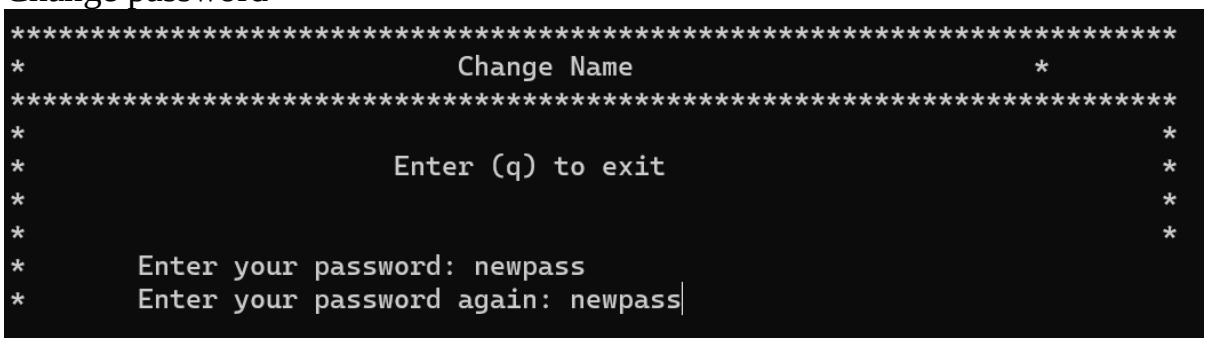
```
**************************************************************************
*                            Withdraw Money                              *
**************************************************************************
*                                                                        *
*                       Enter (q) to exit                                *
*                                                                        *
*                                                                        *
*       Enter an amount to withdraw: 100
```

11. View Profile

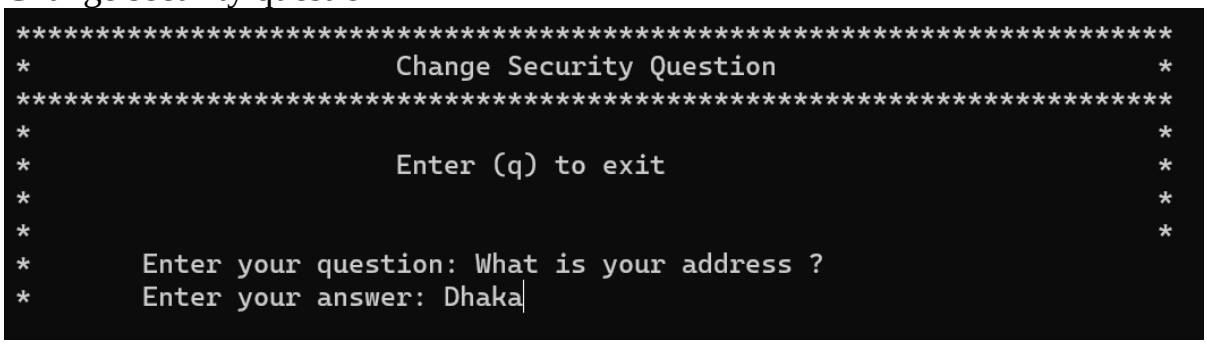
12. Change account name



13. Change password



14. Change security question



Learning Outcome

The Bank Management System is designed for users to create and login to a bank account. Send, deposite, withdraw money without human interaction.

From this assignment, I have learnt to implement a few C concepts in the future projects such as functions, switch statement and do…while statement, arrays, pointers and structures in the program. I have also learnt to create flow charts for explaining the program