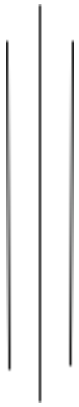
Tribhuvan University
Institute of Engineering
Pulchowk Campus

Department of Electronics and Computer Engineering



A project proposal on

Advanced ATM MACHINE

Bachelor's Degree in Electrical Engineering (BEL)
First Year First Part (I/I)

Submitted To:

Name: Prabesh Sapkota (079BEL055)
Rakesh Karki (079BEL068)
Raunak Mishra (079BEL070)
Sanjaya Limbu (079BEL081)

Department: Electrical Engineering (I/I)

Submitted To:

Department of Electronics and Computer Engineering
Pulchowk Campus, IOE

Table of Contents

.

Introduction:

C programming is a widely used and influential language that was originally developed in the early 1970s by Dennis Ritchie at Bell Labs. It was designed as an evolution of the earlier B programming language and was created to develop the UNIX operating system. A program is a set of instructions or commands that is arranged in a sequence to guide a computer to find a solution for the given problem. Computer programming C is a general purpose programming language that is extremely popular, simple and flexible.

ATM stands for Automated Teller Machine which is a self servicing banking outlet. We can withdraw money, deposit money (as an advanced feature), check our balance.

Doing a project on C programming for an ATM machine can offer several benefits and learning opportunities. It provides practical experience in coding, debugging and problem solving. It allows us to apply the theoretical knowledge that we have gained. This helps us to understand the practical uses of data structures like array, files. Coding a project often comes with its share of bugs and issues. Debugging these problems is a valuable skill that we have developed while working on an ATM project.

This project aims to create a simple working mechanism demonstration of an **ATM** with the help of C programming language and aims to create a great understanding of core C features like different library files, string operations, multi-dimensional arrays, structures and reading, writing operations in file. This project will help us brainstorm the code for the internal working mechanism of an **ATM**, store data in the backend, display and edit the local data files and more.

Problem Statement:

In our project, we have tried to create a basic understanding of ATM machine working features. This project aims to create a basic understanding of how to operate and work with an Automated Teller Machine. Not only understanding ATM machines, we also tried to add an advanced feature of *Depositing Money* to account through ATM Machine, similar to utility payment machines with an addition of a port to insert money and some basic changes to the backend coding, this goal can be achieved. Though we couldn’t demonstrate the hardware integration we tried to make a basic understanding through the backend and coding in C.

Theory:

User defined function:

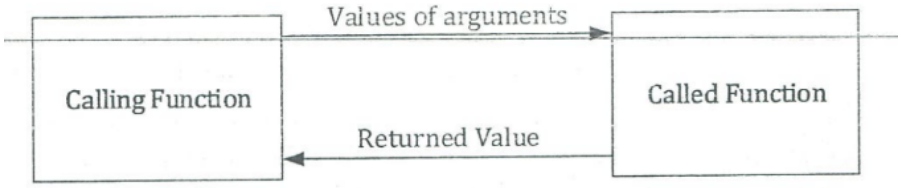
A user-defined function is a modular block of code created by the programmer to perform a specific task. It aids in organizing code, improving reusability, and enhancing program readability. The function consists of a declaration specifying its return type, name, and parameters, followed by a body containing the executable statements. When called from within the program, the function's parameters are passed values, and its body is executed, potentially returning a value to the calling code. Understanding user-defined functions is integral to breaking down complex tasks into manageable components, streamlining code maintenance, and implementing efficient and structured programming practices in the C language.

Syntax:

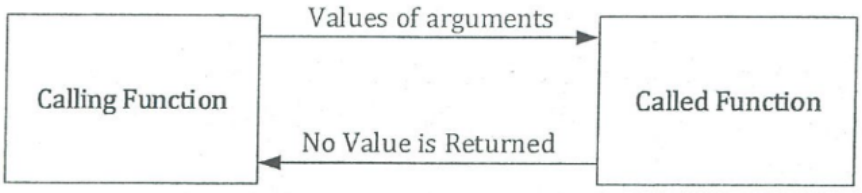
```
return_type function_name(parameters) {
   // Function body (executable statements)
   // ...
   return value; // (if applicable)
}
```

Types:

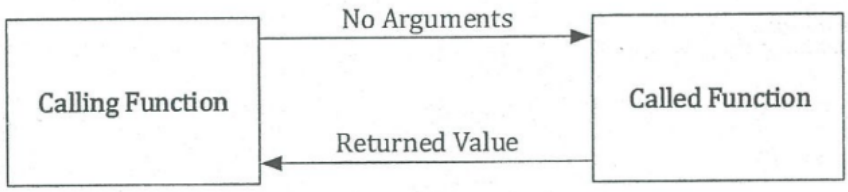
1. **Argument and Value Return:**



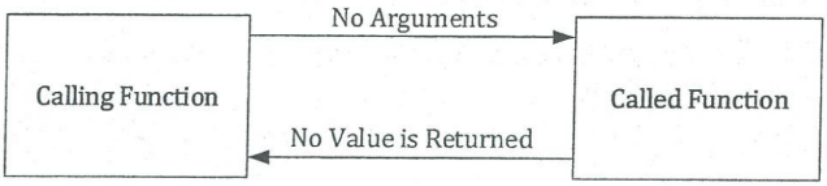
2. **Argument but No Value Return:**



3. **No Argument but Value Return:**



4. **No Argument and No Value Return:**



POINTER:

In C programming, a pointer is a variable that stores the memory address of another variable. Pointers provide a way to indirectly access and manipulate data stored in memory. They are an important concept in C, allowing us to work with dynamic memory allocation, data structures, and function pointers. Here are some key points to understand about pointers in C:

1. *Declaration and Initialization:*
   We declare a pointer variable by specifying the data type it will point to, followed by an *. We can initialize a pointer with the memory address of another variable.
   int x = 10;
   int *ptr;
   ptr = &x;

2. *Dereferencing:*
   Dereferencing a pointer means accessing the value stored at the memory address the pointer points to.
   int value = *ptr;

3. *Pointer Arithmetic:*
   Pointers can be incremented and decremented like normal variables.
   ptr++;

4. *Function Pointers:*
   Pointers can also be used to store the address of a function and allow you to call a function indirectly.

Array:

An array is a sequential collection of elements of the same data type, stored in contiguous memory locations. Array elements are accessed using an index, with the first element at index 0. Arrays provide a way to efficiently store and manage multiple values of the same type, enabling iterative operations and organized data storage. Arrays are declared with a specific size and can be used to represent a wide range of data structures, such as lists, matrices, and strings.

Syntax:

```
data_type array_name[array_size];
```

STRING:

In the context of programming, specifically in the C programming language, a "string" is a sequence of characters that represents textual data. In C, strings are typically implemented as arrays of characters, terminated by a null character ('\0'). Some commonly used string functions in C include strlen(), strcpy(), strcat(), strcmp(), and more.

1. *Declaration of string:*
   We can declare strings in many different ways but the most convenient way to declare a string is declaring it like an array.

   Syntax:

   ```
   char stringName[] = "Hello, World!";
   ```

2. *Accessing Characters in a String:*

   Syntax:

   ```
   char character = stringName[index];
   ```

3. *String Functions:*
   C provides several functions in the standard library for working with strings. Some of the commonly used functions include:

   I. **strlen():**
   Calculates the length of a string.

   II. **strcpy():**
   Copies one string to another.

   III. **strcat():**
   Concatenates (appends) one string to another.

IV. **strcmp():**
Compare two strings.

4. *Looping through Characters:*
You can use loops to iterate through characters in a string.

Syntax:

```
for (int i = 0; stringName[i] != '\0'; i++);
```

NOTE: String are null-terminated (end with a null character, '\0') to indicate the end of the string. Make sure to allocate enough memory for the string, including the null terminator, when declaring and manipulating strings.

Structure:

A structure is a composite data type that groups together variables of different data types under a single name. It allows you to define your own data structure by combining variables of various types into a single unit. This is particularly useful when you need to represent a real-world entity with multiple attributes.

Syntax:

```
struct tag_name {
  type member1;
  type member2;
};
```

DATA FILE IN C:

File handling is a crucial aspect of programming, as it allows you to read and write data to and from files on your computer. In C programming, you can use standard I/O functions to perform file handling operations. Here's an overview of basic file handling operations in C:

1. *Opening a File:*
To work with a file, you need to open it first using the *fopen()* function.

Syntax:

```
filePointer = fopen("filename.txt", "r");
```

2. *Reading from a File:*
You can read data from a file using functions like *fscanf()* or *fgets()*.

Syntax:

```
char buffer[100]; fscanf(filePointer, "%s", buffer);
```

```
fgets(buffer, sizeof(buffer), filePointer);
```

3. *Writing to a File:*
   To write data to a file, you can use functions like *fprintf()* or *fputs()*.

   Syntax:

   ```
   fprintf(filePointer, "Hello, File!\n");
   fputs("Another line of text", filePointer);
   ```

4. *Closing a File:*
   It's important to close a file when you're done working with it using the fclose function.

   Syntax:

   ```
   fclose(filePointer);
   ```

MODES IN FILE HANDLING:

In C programming, file handling modes are used when opening a file to specify the intended type of operations that will be performed on the file. The modes are represented as strings and are passed as the second argument to the *fopen()* function. Here are the commonly used file handling modes:

Mode	Purpose	Syntax
‘r’	Open *text* file for reading only. File must already exist.	*fopen("filename.txt", "r");*
‘w’	Open *text* file for writing only. If a file exists, its content will be destroyed, or a new file will be created if it doesn’t exist already.	*fopen("filename.txt", "w");*
‘a’	Open *text* file for appending only. If a file doesn’t exist, it will be created.	*fopen("filename.txt", "a");*
‘r+’	Open *text* file for reading and writing both. File must already exist.	*fopen("filename.txt", "r+");*
‘b+’	Open *text* file for reading and writing both. If a file exists, its content will be overwritten, or a new file will be created if it doesn’t exist already.	*fopen("filename.txt", "w+");*
‘a+’	Open *text* file for reading and appending both. If a file doesn’t exist, it will be created.	*fopen("filename.txt", "a+");*
‘rb’	Open *binary* file for reading only. File must already exist.	*fopen("filename.txt", "rb");*

‘wb’	Open *binary* file for writing only. If a file exists, its content will be destroyed, or a new file will be created if it doesn’t exist already.	*fopen("filename.txt", "wb");*
‘ab’	Open *binary* file for appending only. If a file doesn’t exist, it will be created.	*fopen("filename.txt", "ab");*
‘r+b’ or ‘rb+’	Open *binary* file for reading and writing both. File must already exist.	*fopen("filename.txt", "r+b");*
‘w+b’ or ‘wb+’	Open a *binary* file for reading and writing both. If a file exists, its content will be overwritten, or a new file will be created if it doesn’t exist already.	*fopen("filename.txt", "w+b");*
‘a+b’ or ‘ab+’	Open a *binary* file for reading and appending both. If a file doesn’t exist, it will be created.	*fopen("filename.txt", "a+b");*

Algorithm:

1. Define Data Structures:

1.1. Define the structures **users** and **user_log_model** to store account details and logged-in user information, respectively.
1.2. Define arrays of users and other necessary variables.

2. Main:

2.1. Clear the screen.
2.2. Take input for user choice from *step 3*.
2.3. If the choice is 0, go to *step 4*.
2.4. Else if choice is 1, go to *step 6*.
2.5. Else print *‘INVALID CHOICE !!! Please enter a valid option’*.

3. Home Screen:

3.1. Display the main menu with options for user login and admin login.
3.2. Take input for user's choice (0 for admin, 1 for user login, other values are invalid).

4. Admin Login:

4.1. Prompt the admin for a *username* and *password*.
4.2. If the credentials are correct, present admin options: create a new user or logout.
4.3. If "Create New User" is chosen, proceed to *step 4*.
4.4. Else return to *step 2*.

5. *Create New User:*
   - 5.1. Get the number of users to create.
   - 5.2. For each user, input first name, last name, account balance, and PIN.
   - 5.3. Generate a unique account number based on a random value and user index.
   - 5.4. Store the user details in the file "student.dat".

6. *User Login:*
   - 6.1. Prompt the user to input their account number.
   - 6.2. Check if the account number is valid and get the corresponding user's PIN.
   - 6.3. If the PIN matches, log in the user and go to *step 7*.

7. *User Choices:*
   - 7.1. Display the user menu with various banking operations.
   - 7.2. Get the user's choice (deposit, withdraw, check balance, etc.).
   - 7.3. Perform the selected operation and update account information.

8. *Deposit Money:*
   - 8.1. Prompt the user for the deposit amount (in multiples of 500).
   - 8.2. Update the logged-in user's account balance with the deposited amount.

9. *Withdraw Money:*
   - 9.1. Prompt the user for the withdrawal amount (in multiples of 500).
   - 9.2. Check if the withdrawal amount is within limits and sufficient funds are available, if not print the error.
   - 9.3. Else withdraw the money and update the logged-in user's account balance after withdrawal.

10. *Fast Cash:*
    - 10.1. Prompt the user to select an amount for fast cash withdrawal.
    - 10.2. Perform the fast cash withdrawal, considering limits and available balance.

11. *Check Balance:*
    - 11.1. Display the logged-in user's current account balance.

12. *Change PIN:*
    - 12.1. Prompt the user to enter a new 5-digit PIN.
    - 12.2. Update the logged-in user's PIN.

13. Update:

13.1. Update the account information (name, number, balance, PIN) after performing operations.

13.2. Store the updated information in a temporary file and replace the original file with the updated one.

14. Repeat:

14.1. Allow the user to continue with additional operations or log out and return to *step 2*.

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
struct users{
 char accHolderName[40],accNumber[30],accPassword[10],accBalance[20];
};
struct user_log_model{
 char accHolderName[40],accNumber[30],accPassword[10];
 unsigned long int accBalance;
}logged_user;
struct users u[100], copyUser;
int homeScreen();
int adminlogin();
int userlogin();
int createNewUser();
int checkAccountNumber(char aN[30]);
int checkPassword(int p, char accountPin[30]);
int userChoices();
void deposit_money();
void withdraw_money();
void fast_cash();
void check_balance();
void change_pin();
void update();
int main(){
 int choice;
 clrscr();
```

```
 choice=homeScreen();
 if(choice==0){
  adminlogin();
 }else if(choice==1){
  userlogin();
 }else{
  printf("\nINVALID CHOICE !!! Please enter a valid option.");
 }
 getch();
 return 0;
}
int homeScreen(){
 int c;
 printf(".....Welcome to Automated Banking System.....");
 printf("\nPlease enter the code corresponding to the operation you want to
perform.\n1.\tLogin for Users\n\n0.\tAdmin Login\n");
 fflush(stdin);
 scanf("%d",&c);
 return c;
}
int adminlogin(){
 char username[30],password[30];
 int in,ad_ch;
 printf("\nPlease enter your username:\t");
 fflush(stdin);
 scanf("%s",username);
 if(strcmp(username,"admin")==0){
  printf("\nPlease enter your password:\t");
  fflush(stdin);
  char charc;
  while((charc=getch())!=13){
   printf("*");
   password[in]=charc;
   in++;
  }
  password[in]='\0';
  if(strcmp(password,"admin@2023#")==0){
   printf("\nPlease enter the code corresponding to the action you want to
perform:\n1.\tCreate New User\n2.\tLogout\n");
   fflush(stdin);
```

```
   scanf("%d",&ad_ch);
   if(ad_ch==1){
    createNewUser();
   }else if(ad_ch==2){
    clrscr();
    main();
   }else{
    printf("\nINVALID CHOICE !!! Please enter a valid option.");
   }
  }else{
   printf("\nINVALID PASSWORD... Please enter a valid password.\n");
   return 1;
  }
 }else{
  printf("\nINVALID USERNAME... Please enter a valid username.\n");
  return 1;
 }
 return 0;
}
int createNewUser(){
 char f_name[40],l_name[20],pass[10];
 FILE *fp;
 int n;
 printf("\nEnter the number of users you want to create:\t");
 fflush(stdin);
 scanf("%d",&n);
 fp=fopen("student.dat","ab");
 if(fp==NULL){
  printf("\nAn unexpected error occurred... Please try again.");
  exit(1);
 }
 else{
  for(int i=0;i<n;i++){
   printf("\nFor User[%d]:",i+1);
   printf("\nEnter first name:\t");
   fflush(stdin);
   gets(f_name);
   printf("\nEnter last name:\t");
   fflush(stdin);
   gets(l_name);
```

```
strcat(f_name," ");
strcat(f_name,l_name);
strcpy(u[i].accHolderName,f_name);
randomize();
int accN=random(1000000),accNcopy=accN,accsize=0,accNu[6];
char accNum[30],bankcode[30];
while(accN>0){
 accN%10;
 accsize++;
 accN/=10;
}
for(int v=0;v<6-accsize;v++){
 accNu[v]=0;
}
for(int j=6-accsize;j<6;j++){
 accNu[j]=accNcopy%10;
 accNcopy/=10;
}
int index=0;
for(int l=0;l<6;l++){
 index+=sprintf(&accNum[index],"%d",accNu[l]);
}
strcpy(bankcode,"00100501");
strcat(bankcode,accNum);
strcpy(u[i].accNumber,bankcode);
printf("\nEnter Account Balance:\t");
fflush(stdin);
scanf("%s",u[i].accBalance);
printf("\nEnter a 5-digit pin for this account:\t");
fflush(stdin);
char charc1,hideninput[10];
int in1=0;
while((charc1=getch())!=13){
 printf("*");
 hideninput[in1]=charc1;
 in1++;
}
hideninput[in1]='\0';
strcpy(u[i].accPassword,hideninput);
```

```
   printf("\nUser created successfully...\nName:\t%s\nAccount Number:\t%s\nAccount
Balance:\t%s\nPassword:\t%s\n",u[i].accHolderName,u[i].accNumber,u[i].accBalance,u[i
].accPassword);
  }
  fwrite(&u,sizeof(struct users),100,fp);
  fclose(fp);
 }
 return 0;
}
int userlogin(){
 char accountNumber[30];
 int accComp;
 printf("\nPlease Enter your account number:\t");
 fflush(stdin);
 scanf("%s",accountNumber);
 accComp=checkAccountNumber(accountNumber);
 if(accComp==0){
  printf("\nSorry you couldn't be logged in...");
 }else if(accComp==1){
  printf("\nYou're successfully logged in!!!");
  userChoices();
 }else{
  main();
  printf("\nAn unexpected error occurred. Please try again...");
 }
 return 0;
}
int checkAccountNumber(char aN[30]){
 FILE *fp1;
 char accountPin[30];
 int log=0,k;
 struct users z[100];
 fp1=fopen("student.dat","r");
 if(fp1==NULL){
  printf("\nAn unexpected error occurred... Please try again.");
  exit(1);
 }
 else{
  for(k=0;k<100;k++){
   do{
```

```
    fread(&u,sizeof(struct users),1,fp1);
    if(strcmp(u[k].accNumber,aN)==0){
     printf("\nPlease Enter your account pin:\t");
     fflush(stdin);
     char charc2,hideninput2[10];
     int in2=0;
     while((charc2=getch())!=13){
      printf("*");
      hideninput2[in2]=charc2;
      in2++;
     }
     hideninput2[in2]='\0';
     strcpy(accountPin,hideninput2);
     if(strcmp(u[k].accPassword,accountPin)==0){
      strcpy(logged_user.accHolderName,u[k].accHolderName);
      strcpy(logged_user.accNumber,u[k].accNumber);
      logged_user.accBalance=atoi(u[k].accBalance);
      strcpy(logged_user.accPassword,u[k].accPassword);
      log=1;
     }
     break;
    }
   }while(!feof(fp1));
   break;
  }
 }
 fclose(fp1);
 return log;
}
int userChoices(){
 int user_ch;
 strupr(logged_user.accHolderName);
 printf("\n\n\tHello %s",logged_user.accHolderName);
 printf("\n\tPlease enter the code corresponding to the action you want to
perform:\n\t1.\tDeposit Money\n\t2.\tWithdraw Money\n\t3.\tFast Cash\n\t4.\tCheck
Balance\n\t5.\tChange Pin\n\t0.\tCancel\n");
 scanf("%d",&user_ch);
 if(user_ch==0){
  main();
 }else if(user_ch==1){
```

```
 deposit_money();
 update();
}else if(user_ch==2){
 withdraw_money();
 update();
}else if(user_ch==3){
 fast_cash();
 update();
}else if(user_ch==4){
 check_balance();
}else if(user_ch==5){
 change_pin();
 update();
}else{
 main();
}
getch();
return 1;
}
void deposit_money(){
 unsigned long int depo_amt;
 printf("\nPlease enter the amount you want to deposit (in multiples of 500 only)\n");
 fflush(stdin);
 scanf("%lu",&depo_amt);
 if((depo_amt%500)!=0){
  printf("\nPlease enter amount in multiple of 500.");
 }else{
  logged_user.accBalance+=depo_amt;
  printf("\nPlease insert %lu cash.",depo_amt);
 }
}
void withdraw_money(){
 unsigned long int draw_amt;
 printf("\nPlease enter the amount you want to withdraw (in multiples of 500 only)\n");
 fflush(stdin);
 scanf("%lu",&draw_amt);
 if(draw_amt>logged_user.accBalance){
  printf("\nInsufficient Balance!!!");
 }else if(draw_amt>25000){
  printf("\nMaximum Withdraw limit exceeded...");
```

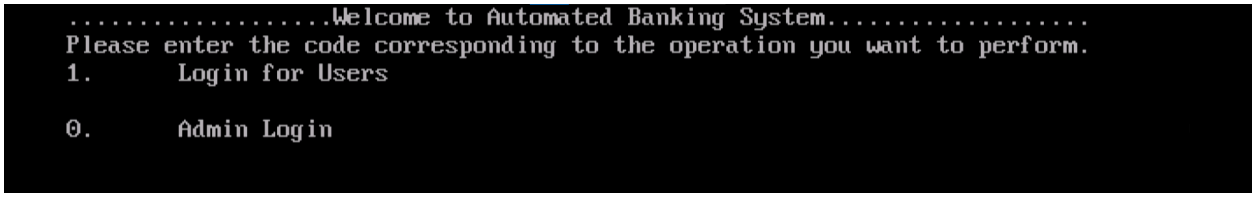
```
 }else if((draw_amt%500)!=0){
  printf("\nPlease enter amount in multiple of 500.");
 }else{
  logged_user.accBalance-=draw_amt;
  printf("\nPlease collect your cash and card.");
 }
}
void fast_cash(){
 int fast_ch;
 unsigned long int fast_draw_amt;
 printf("\nPlease enter the code corresponding to the amount you want to withdraw:\n1.
1000\t2. 2000\n3. 4000\t4. 5000\n5. 10000\t6.15000\n7.20000\t8.25000\n");
 fflush(stdin);
 scanf("%d",&fast_ch);
 if(fast_ch==1){
  fast_draw_amt=1000;
 }else if(fast_ch==2){
  fast_draw_amt=2000;
 }else if(fast_ch==3){
  fast_draw_amt=4000;
 }else if(fast_ch==4){
  fast_draw_amt=5000;
 }else if(fast_ch==5){
  fast_draw_amt=10000;
 }else if(fast_ch==6){
  fast_draw_amt=15000;
 }else if(fast_ch==7){
  fast_draw_amt=20000;
 }else if(fast_ch==8){
  fast_draw_amt=25000;
 }else{
  main();
 }
 if(fast_draw_amt>logged_user.accBalance){
  printf("\nInsufficient Balance!!!");
 }else{
  logged_user.accBalance-=fast_draw_amt;
  printf("\nPlease collect your cash and card.");
 }
}
```

```
void check_balance(){
 printf("\nYour current balance is: %d",logged_user.accBalance);
}
void change_pin(){
 printf("\nPlease enter a new pin of 5-digit:\t");
 fflush(stdin);
 char charc3,hideninput3[10];
 int in3=0;
 while((charc3=getch())!=13){
  printf("*");
  hideninput3[in3]=charc3;
  in3++;
 }
 hideninput3[in3]='\0';
 strcpy(logged_user.accPassword,hideninput3);
}
void update(){
 unsigned long int ab=logged_user.accBalance,abrev=0;
 while(ab!=0){
  abrev=abrev*10;
  abrev=abrev+ab%10;
  ab=ab/10;
 }
 unsigned long int AccNum=abrev,AccNCopy=abrev,AccNu[8];
 int AccSize=0;
 char AccBal[8];
 while(AccNum>0){
  AccNum%10;
  AccSize++;
  AccNum/=10;
 }
 for(int g=0;g<8-AccSize;g++){
  AccNu[g]=0;
 }
 for(int d=8-AccSize;d<8;d++){
  AccNu[d]=AccNCopy%10;
  AccNCopy/=10;
 }
 int indexLog=0;
 for(int q=0;q<8;q++){
```

```
  indexLog+=sprintf(&AccBal[indexLog],"%lu",AccNu[q]);
 }
 strcpy(copyUser.accHolderName,logged_user.accHolderName);
 strcpy(copyUser.accNumber,logged_user.accNumber);
 strcpy(copyUser.accBalance,AccBal);
 strcpy(copyUser.accPassword,logged_user.accPassword);
 //printf("Current User:\nName:\t%s\nAccount
Number:\t%s\nBalance:\t%s\nPassword:\t%s",copyUser.accHolderName,copyUser.accN
umber,copyUser.accBalance,copyUser.accPassword);
 FILE *fpOrg,*fpCopy;
 fpOrg=fopen("student.dat","rb");
 fpCopy=fopen("stu_copy.dat","ab");
 if(fpOrg==NULL || fpCopy==NULL){
  printf("\nAn unexpected error occurred, please try again...");
  exit(1);
 }else{
  fread(&u,sizeof(struct users),100,fpOrg);
  for(int w=0;w<100;w++){
   if(strcmp(u[w].accNumber,logged_user.accNumber)==0){
    strcpy(u[w].accHolderName,copyUser.accHolderName);
    strcpy(u[w].accNumber,copyUser.accNumber);
    strcpy(u[w].accBalance,copyUser.accBalance);
    strcpy(u[w].accPassword,copyUser.accPassword);
   }
  }
  fwrite(&u,sizeof(struct users),100,fpCopy);
 }
 fclose(fpOrg);
 fclose(fpCopy);
 remove("student.dat");
 rename("stu_copy.dat","student.dat");
}
```

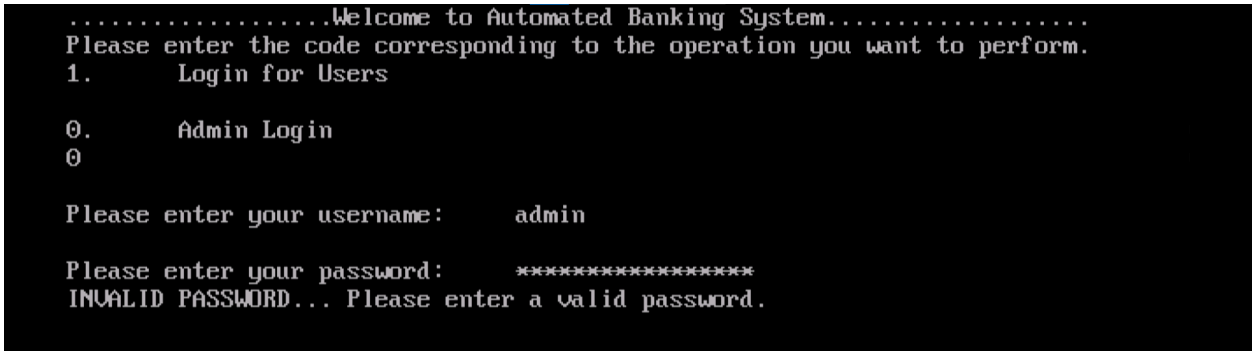
Output:

Home Screen:

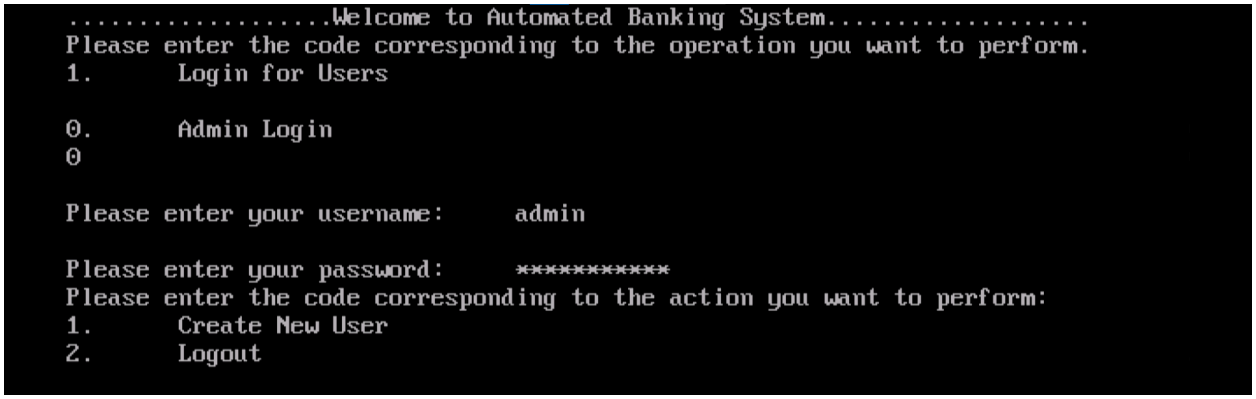


Admin Login (Selection of 0) on Home Screen:

- On Incorrect username or password entered:



- On Successful Login



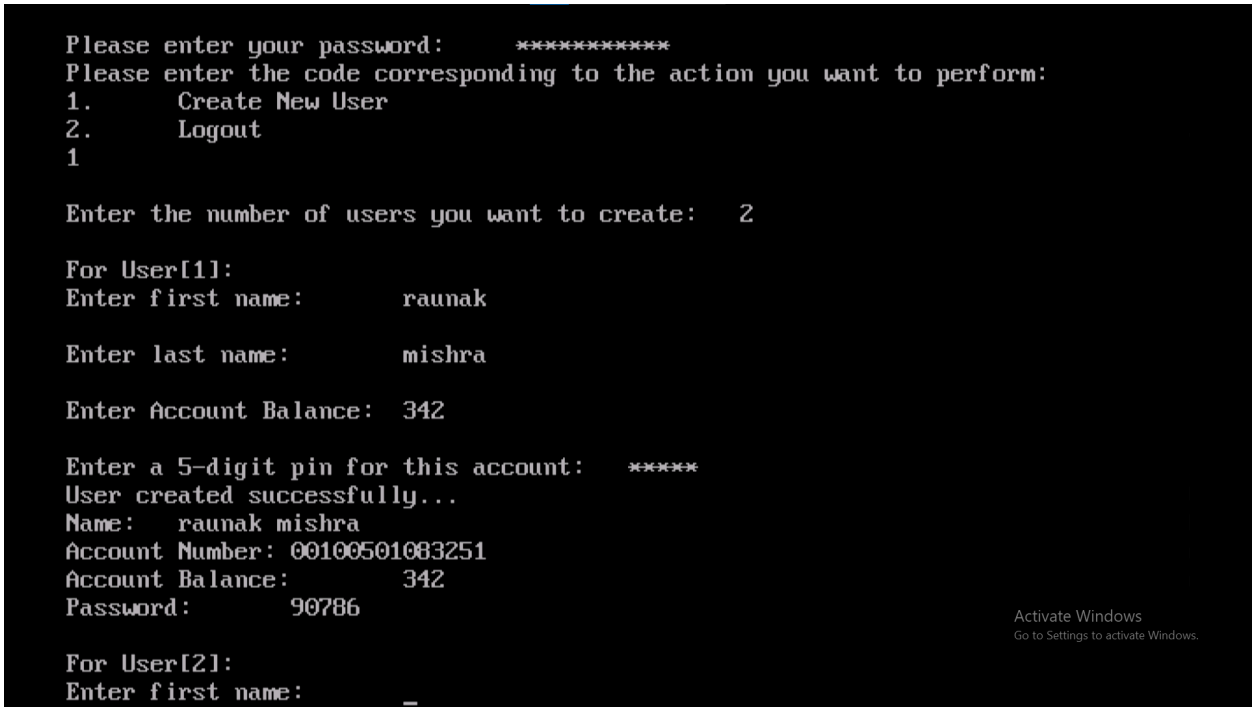
➢ Selection of Choice 2

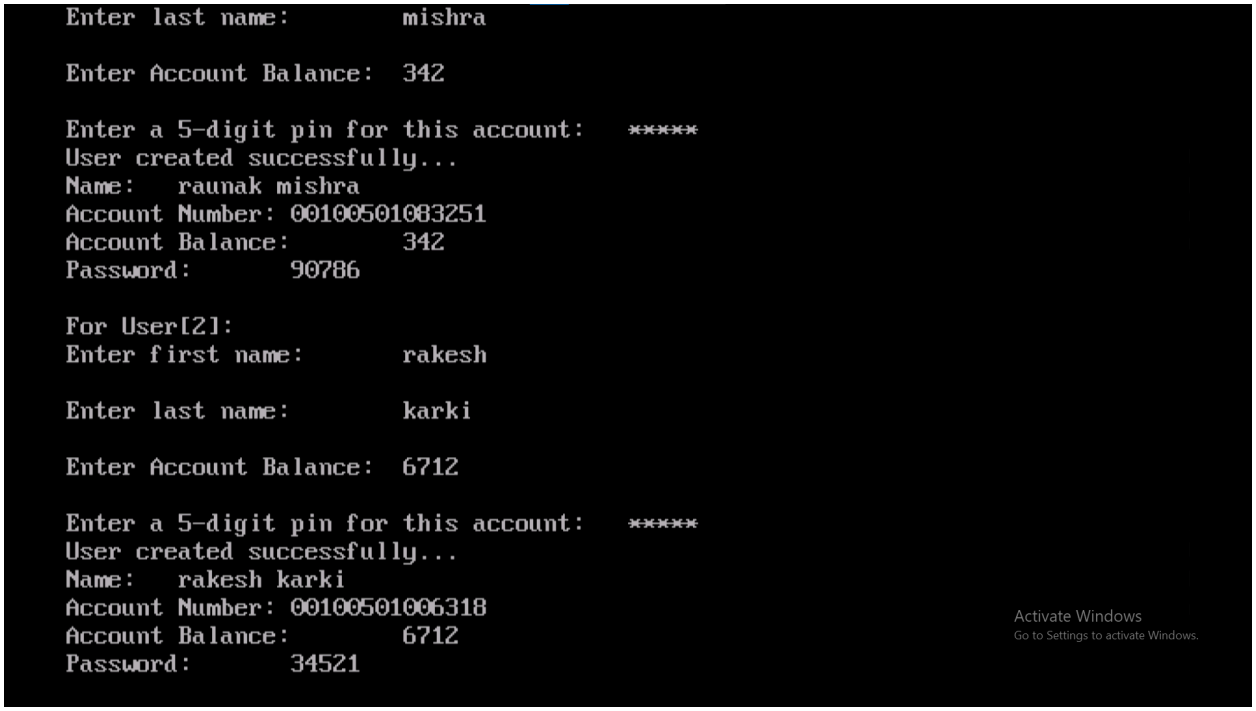
```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
0

Please enter your username:     admin

Please enter your password:     ***********
Please enter the code corresponding to the action you want to perform:
1.      Create New User
2.      Logout
2
```

Back to Home Screen

```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
-
```

➢ Selection of Choice 1





User Login (Selection of 1) on Home Screen:

```
...................Welcome to Automated Banking System...................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:
```

- On Incorrect username entered:

```
...................Welcome to Automated Banking System...................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501009867

Sorry you couldn't be logged in..._
```

- On Incorrect password entered:

```
...................Welcome to Automated Banking System...................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501006318

Please Enter your account pin:  ********
Sorry you couldn't be logged in..._
```

- On Successful Login:

```
...................Welcome to Automated Banking System...................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501006318

Please Enter your account pin:  ******
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel

_
```

➢ Selection of Choice 0

```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501000186

Please Enter your account pin:  ******
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
0
```

Back to Home Screen

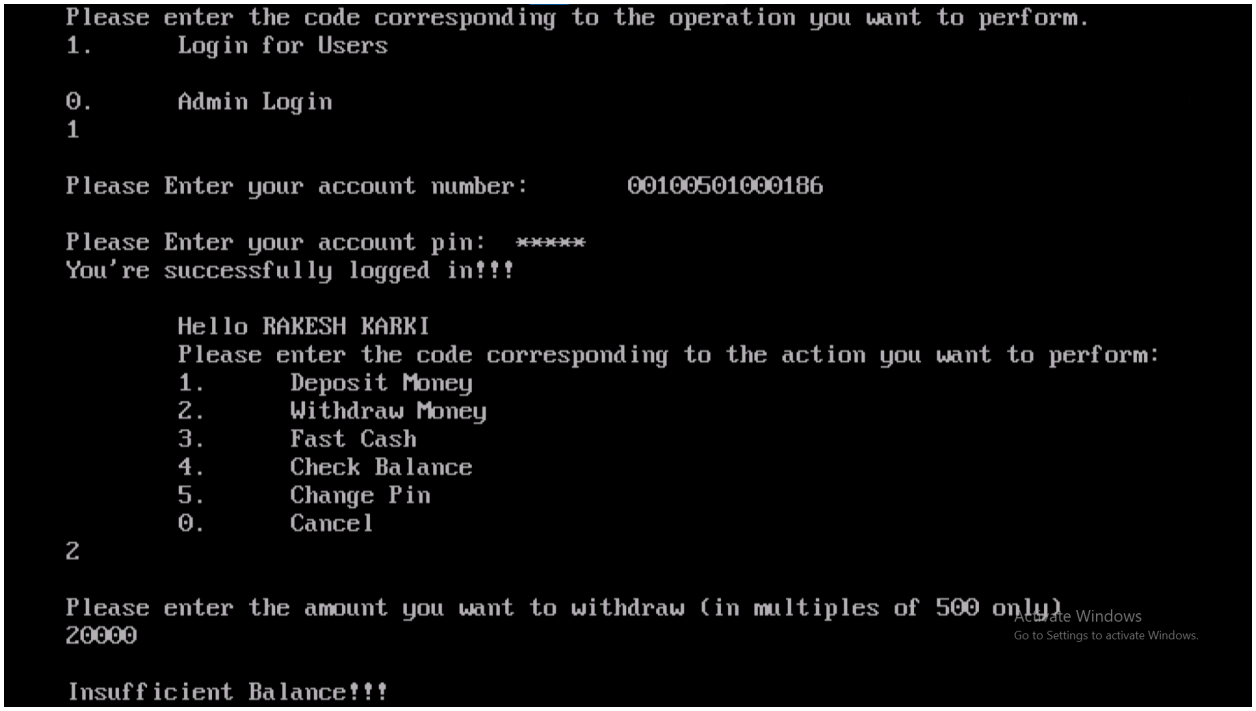
```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
_
```

➢ Selection of Choice 1

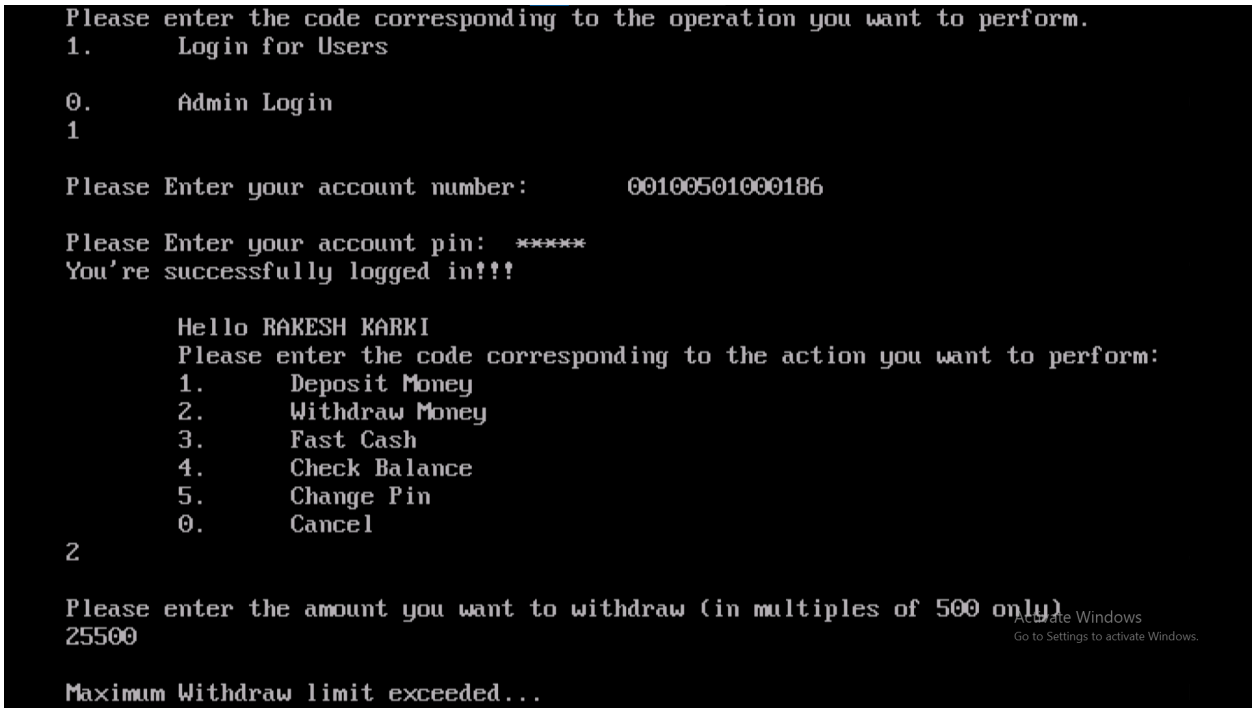
```
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501006318

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
1

Please enter the amount you want to deposit (in multiples of 500 only)
1000

Please insert 1000 cash.
```

➢ Selection of Choice 2

- Case of Insufficient Balance:



- Case of maximum withdraw limit reached:



- Case of amount non-multiple of 500:

```
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501000186

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
2

Please enter the amount you want to withdraw (in multiples of 500 only)
5643

Please enter amount in multiple of 500.
```


- Case of All Positive Conditions:

```
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501000186

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
2

Please enter the amount you want to withdraw (in multiples of 500 only)
2000

Please collect your cash and card.
```


➢ Selection of Choice 3

```
1

Please Enter your account number:       00100501000186

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
3

Please enter the code corresponding to the amount you want to withdraw:
1. 1000 2. 2000
3. 4000 4. 5000
5. 10000        6.15000
7.20000 8.25000
4

Please collect your cash and card._
```


➢ Selection of Choice 4

```
.................Welcome to Automated Banking System.................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501006318

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
4

Your current balance is: 7712
```


➢ Selection of Choice 5

```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501006318

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
5

Please enter a new pin of 5-digit:      *****_
```


➢ Invalid Input

```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
1

Please Enter your account number:       00100501000186

Please Enter your account pin:  *****
You're successfully logged in!!!

        Hello RAKESH KARKI
        Please enter the code corresponding to the action you want to perform:
        1.      Deposit Money
        2.      Withdraw Money
        3.      Fast Cash
        4.      Check Balance
        5.      Change Pin
        0.      Cancel
9
```

Back to Home Screen

```
..................Welcome to Automated Banking System..................
Please enter the code corresponding to the operation you want to perform.
1.      Login for Users

0.      Admin Login
```

Discussion and Conclusion:

In the project, we demonstrated the working mechanism of an ATM Machine and learnt about its backend processes with the help of C. We learnt about different features of C through this program. We learnt about different library/header files in C, operations on string like string concatenation, string comparison, string copy, converting a string to uppercase, array and its operations like assigning value to an array index, reversing elements of an array and mathematical operations on array elements, structures and working of structures with both homogeneous and heterogeneous data types and file operations like creating/ reading/ appending/ removing/ renaming of files, reading or writing complete structure and array direct into file without any loop, and obviously different forms of error handling. We also learnt about different types of user-defined functions, i.e, Argument and Value Return Function, No Argument But Value Return Function, Argument But No Value Return Function, No Argument and No Value Return Function and all these four different kind of functions as per need. We also learnt to hide the user input in case of sensitive fields and display '*' in case of actual input value.

Hence, we've had a good knowledge of user defined functions, arrays, strings, structures and file operations in C. Along with that we also learnt about different header files in C, understood the concept of pointers, memory storage for different data types, arrays, and structures, and flushing the former inputs wherever necessary.