DATA STRUCTURE USING C

BANK MANAGEMENT SYSTEM USING LINKED LIST

Guided by:
Prof. V. M. Bande.

Submitted by:
Kaushal Khachane (33)
Abhishek Patil (49)

Contents

Literature Review

- Bank Management System using c this project is developed using linked list. Linked list is linear data structure.

- Dynamic memory allocation make linked list operation efficient. We can store customer data upto memory becomes available. Linked list stored collection of data.

- Various operation performed on linked list are insertion, deletion, traversing. We can use such operation in our project. By using such function we implement bank management system project.

Abstract

- The Bank Account Management System is an application for maintaining a person's account in a bank.
- In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System.
- In this Project user can Create Account, Modify Account, Manage Transaction, Delete Account, Search Existing Account and Display All the Account In Bank.
- The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly.

Introduction

1. We developed **Microproject in C bank management system using linked list** during our third semester. This project is focused on customer account services in bank.

2. Here, we can create a new account, update information of an existing account, view and manage transactions, check the details of an existing account, remove existing account and view customers' list.

3. Overall, with this project, you can perform banking activities like in a Real bank.

4. The project code for this project is around 470 lines. The project code is password protected (password is dsuproject).

Data structure used

In this project we use singly linked list. A linked list a data structure which contain two field.

1. Data field
2. Pointer to next node

Head pointer contain the address of the first node and the last node contain NULL to indicate end of the list.

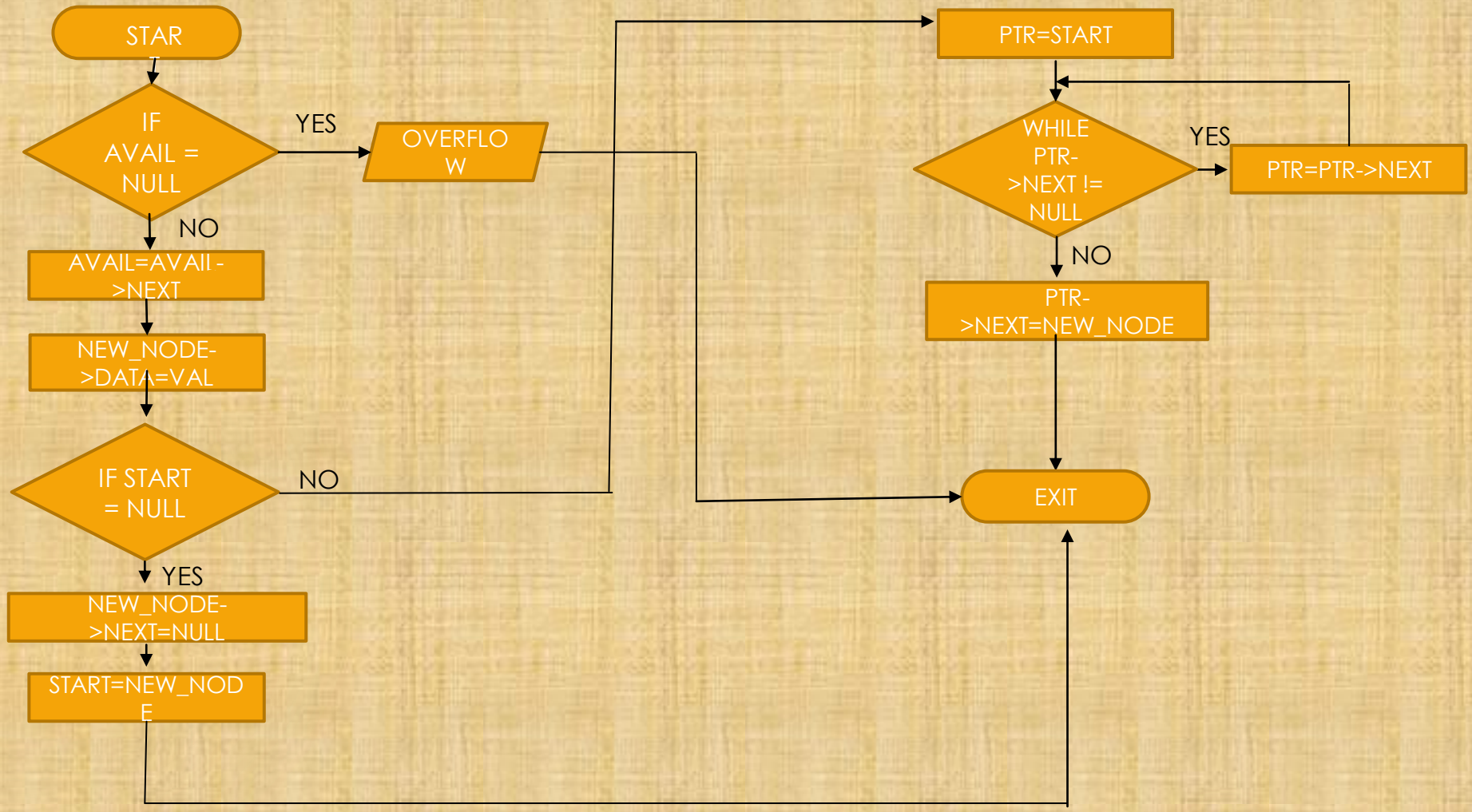
Node Creation:-

```
Struct node{
    int data;
    struct node *next;
};
Struct node *Head, *Ptr;
New node = (struct node *)malloc(size of(struct node));
```

Algorithm To Insert Record Into Bank

Step 1: START
Step 2: IF AVAIL = NULL
Write OVERFLOW
Go to step 15
Step 3: SET NEW_NODE = AVAIL
Step 4: SET AVAIL = AVAIL->NEXT
Step 5: SET NEW_NODE->DATA = VAL
Step 6: IF START = NULL
Step 7: NEW_NODE->NEXT = NULL
Step 8: START = NEWNODE
Go To Step 15
Step 9: ELSE
Step 10: PTR = START
Step 11: WHILE PTR->NEXT != NULL
Step 12: PTR = PTR->NEXT
[END OF LOOP]
Step 13: PTR->NEXT = NEW_NODE
Step 14: NEW_NODE->NEXT = NULL
[END OF IF]
Step 15: EXIT

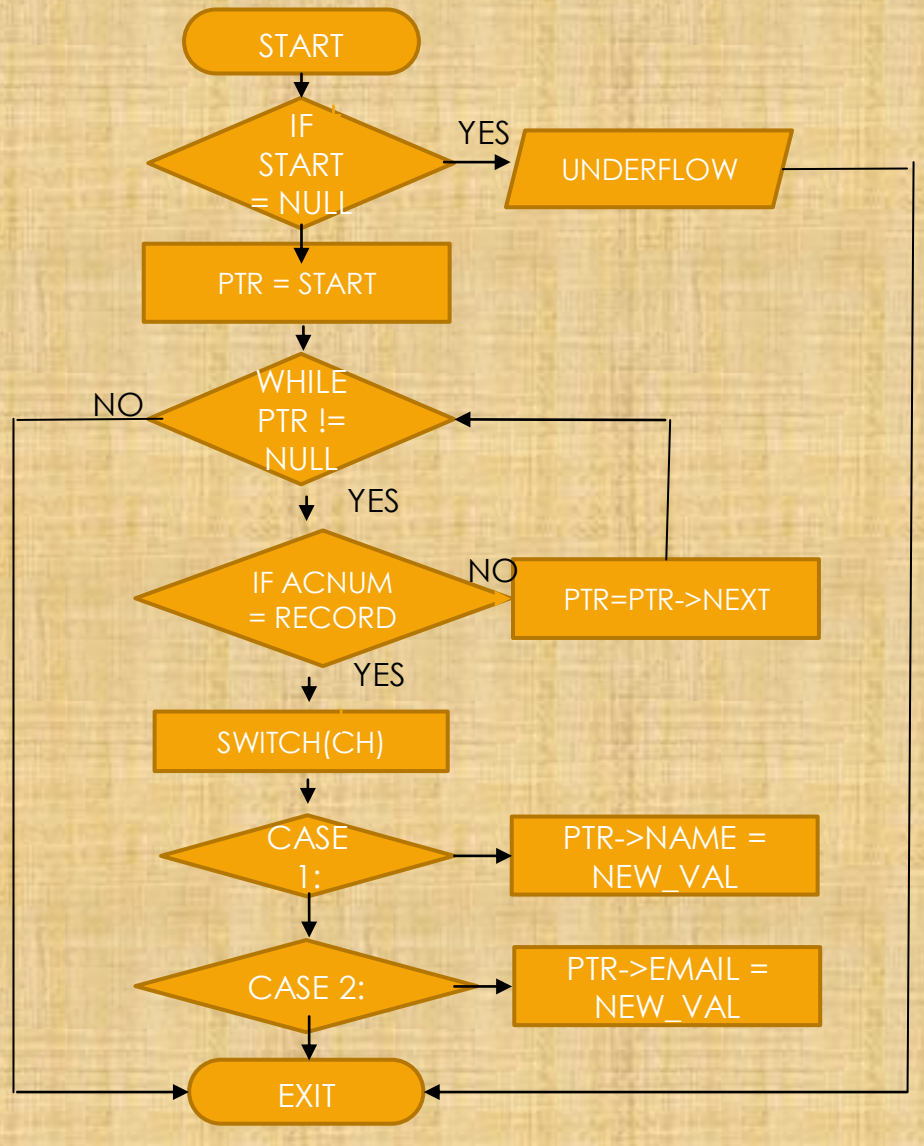
Flowchart To Insert Record Into Bank



Algorithm To Update Details Of Exiting Account

Step 1: START
Step 2: IF START = NULL
Write UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
Go To Step 14
Step 4: ELSE
Step 5: PTR = START
Step 6: WHILE PTR != NULL
Step 7: IF AcNum = Record
Step 8: Switch(ch)
Step 9: Case 1: PTR->NAME = NEW_VAL BREAK
Step 10: Case 2: PTR->ADDRESS = NEW_VAL BREAK;
Step 11: Default: TRY AGAIN
Step 12 ELSE
Step 14: PTR = PTR->NEXT
[END OF IF]
[END OF LOOP]
[END OF IF]
Step 14: EXIT

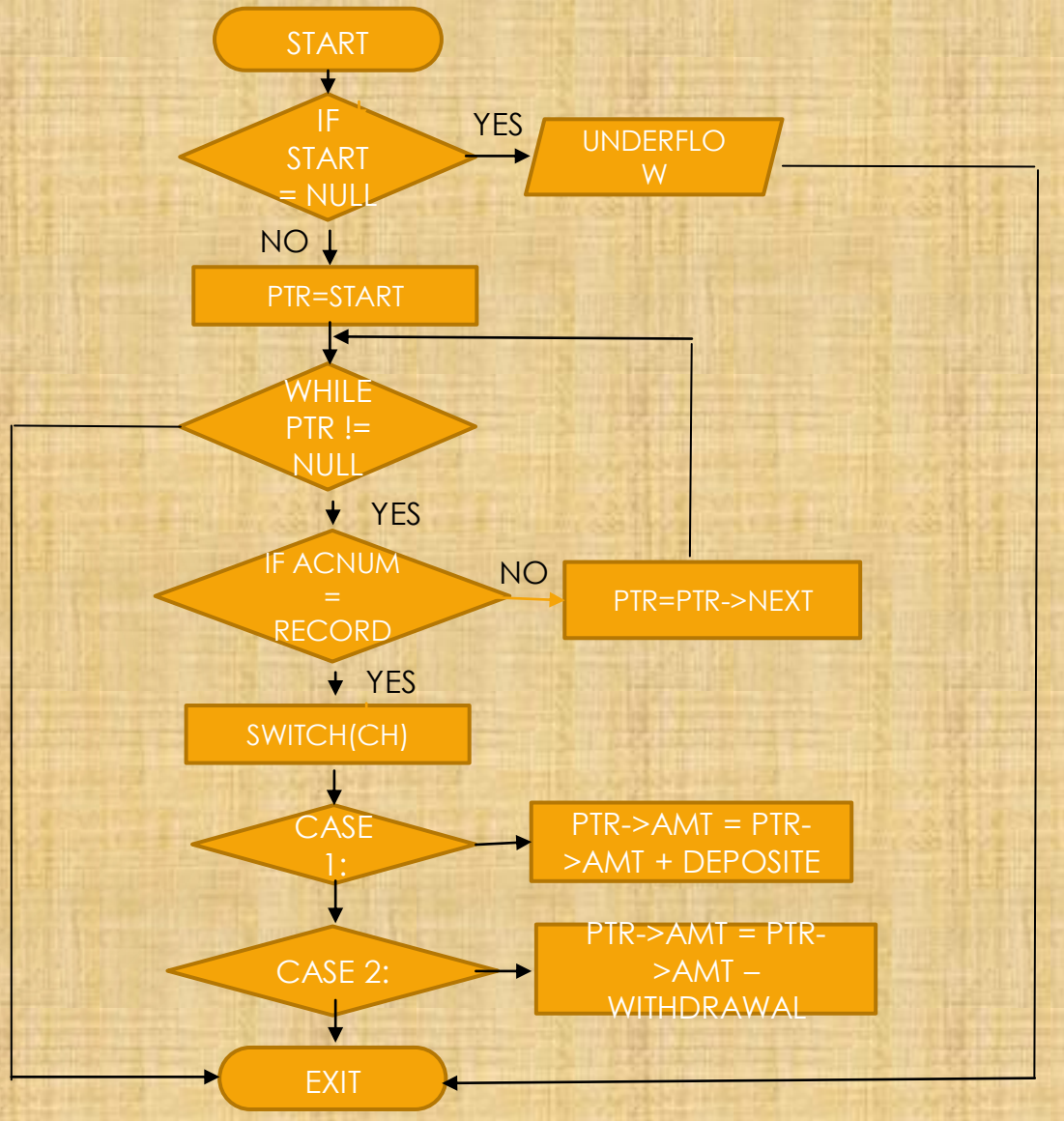
Flowchart To Update Details Of Exiting Account



Algorithm To Manage Transaction

Step 1: START
Step 2: IF START = NULL
Write UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
Go To Step 12
Step 3: ELSE
Step 4: PTR = START
Step 5: WHILE PTR != NULL
Step 6: IF AcNum = Record
Step 7: Switch(ch)
Step 8: Case 1 = PTR->AMT = PTR->AMT + DEPOSITE
Step 9: Case 2 = PTR->AMT = PTR->AMT – WITHDRAWAL
Default = TRY AGAIN
Step 10: ELSE
Step 11: PTR = PTR->NEXT
[END OF IF]
[END OF LOOP]
[END OF IF]
Step 12: EXIT

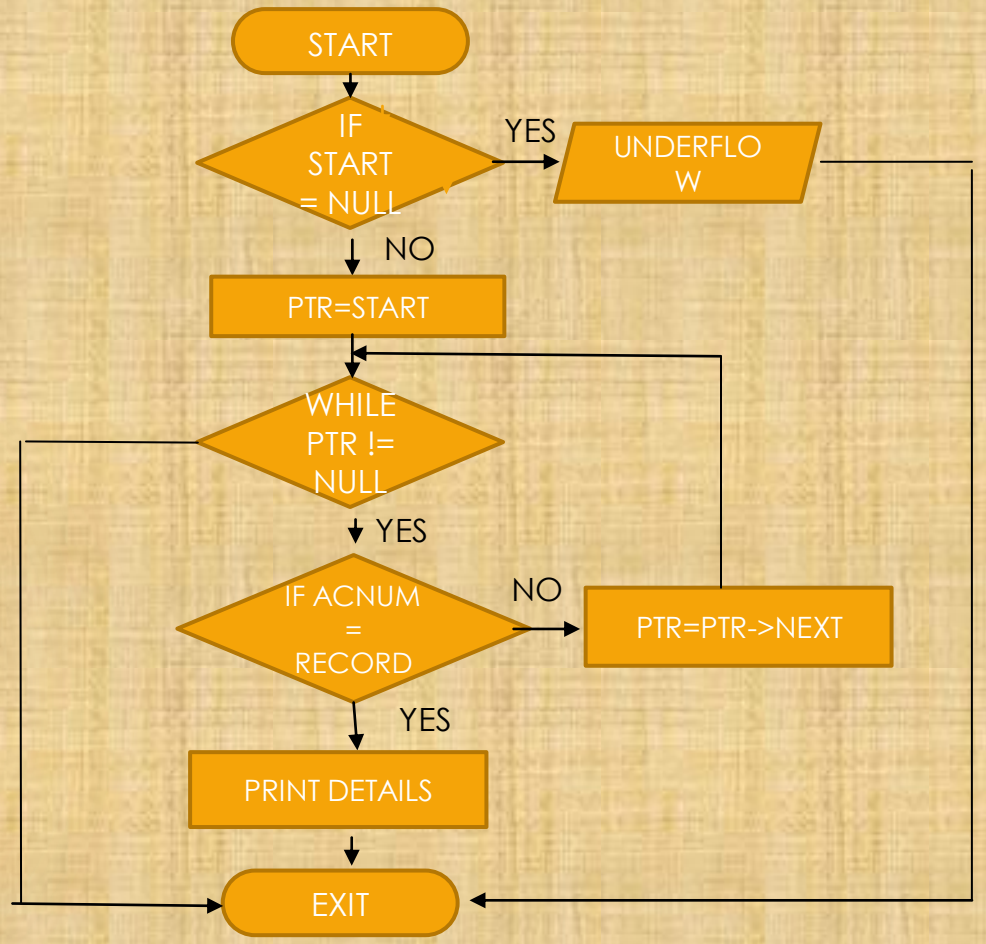
Flowchart To Manage Transaction



Algorithm To Check Details of Existing Account

Step 1: START
Step 2: IF START = NULL
Write UNDERFLOW (THERE IS NO ACCOUNT IN BANK)
Go To Step 10
Step 3: ELSE
Step 4: PTR = START
Step 5: WHILE PTR != NULL
Step 6: IF AcNum = Record
Step 7: PRINT DETAILS
Step 8: ELSE
Step 9: PTR = PTR->NEXT
[END OF IF]
[END OF LOOP]
[END OF IF]
Step 10: EXIT

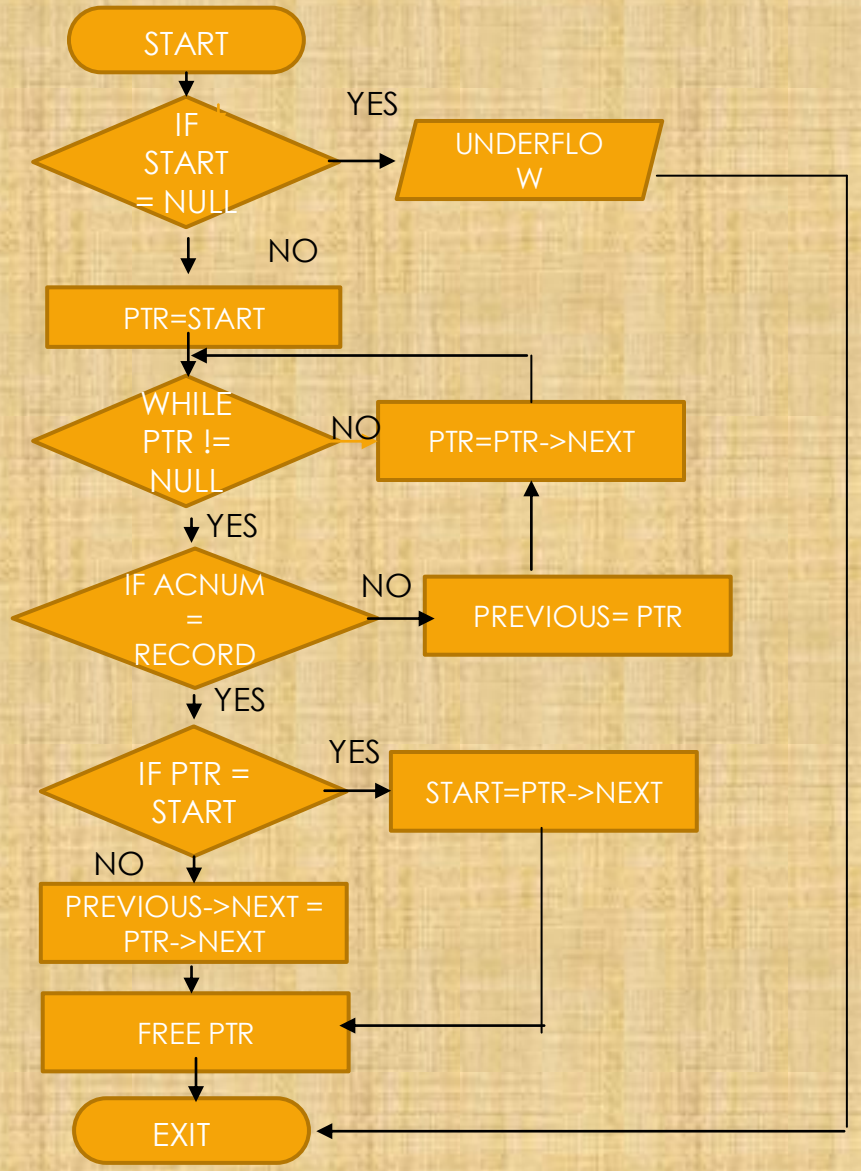
Flowchart To Check Details of Existing Account



Algorithm To Remove Existing Account

```
Step 1: START
Step 2: IF START = NULL
                Write UNDERFLOW (THERE IS NO BANK ACCOUNT)
          [END OF IF]
Step 3: PTR = START
Step 4:   WHILE PTR != NULL
Step 5:         IF AcNum = Record
Step 6:               IF PTR = START
Step 7:                     START = PTR->NEXT
Step 8:               ELSE
Step 9:                     PREVIOUS->NEXT = PTR->NEXT
Step 10:              FREE PTR
                      [END OF IF]
Step 11:        ELSE
Step 12:              PREVIOUS = PTR
Step 13:              PTR = PTR->NEXT
                [END OF IF]
            [END OF LOOP]
          [END OF IF]
Step 14: EXIT
```

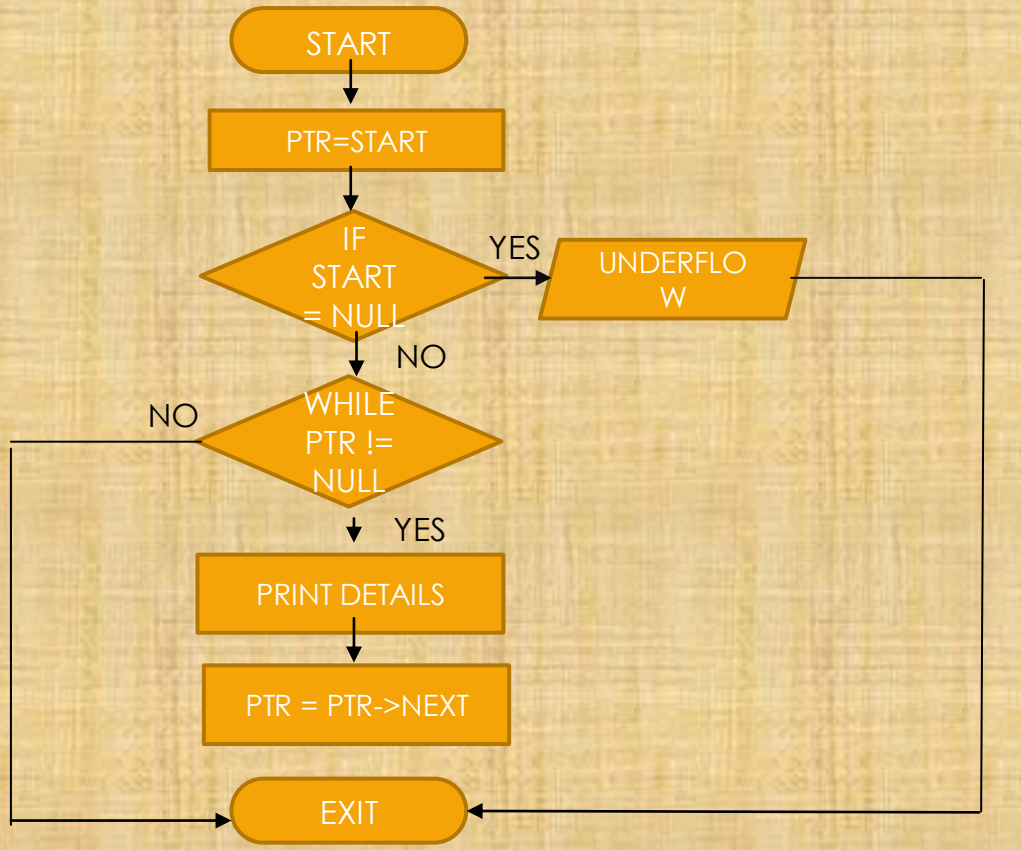
Flowchart To Remove Existing Account



Algorithm To Display All Record

Step 1: START
Step 2: PTR = START
Step 3: IF START = NULL
Write UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
GO TO STEP 8
[END OF IF]
Step 4: ELSE
Step 5: DO WHILE PTR !=NULL
Step 6: PRINT DETAILS
Step 7: PTR = PTR->NEXT
[END OF DO WHILE]
[END OF IF]
Step 8: EXIT

Flowchart To Display All Record



Micro Project Outputs

1. Menu

```
CUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM


    WELCOME TO THE MAIN MENU

    ============================
    1. Create new account
    2. Update information of existing account
    3. For tracsacctions
    4. Check a details of existing account
    5. Remove existing account
    6. view customer's list
    7. Exit
    ============================
```

2. Create Account

```
                ADD RECORD

        Enter today's date (dd mm yyyy):14 01 2021
        Enter account number: 10
        Enter Name: Bhupesh
        Enter the date of birth (dd mm yyyy):22 01 2003
        Enter the phone Number: 87459547819
        Enter the Address: Mumbai
        Enter the amount to deposit:$41000
        Type of account:
        #Saving
        #Current
        #Fixed1(for 1 year)
        #Fixed2(for 2 years)
        #Fixed3(for 3 years)
        Enter your choice:#Current

    Account added succesfully...
```

3. Modify Account Record

```
                MODIFY RECORD

AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1               Kaushal     550000.000000       #Saving
2               Abhishek    50000.000000        #Current
3               Kaustubh    70000.000000        #Saving
4               Shantanu    45000.000000        #Current
5               Pavan       45000.000000        #Fixed1
6               Anurag      51000.000000        #Fixed2
7               Rajat       78000.000000        #Saving
8               Shubham     45100.000000        #Current
9               Amarbir     90000.000000        #Fixed3
10              Bhupesh     41000.000000        #Current
-------------------------------------------------------

        Enter Account No.. To Modify: 7

        1 - To Change Name
        2 - To Change Address

        Enter your choice: 1
        Enter New Name: Gaurav

                Name Updated to Gaurav
```

4. Check Details Of Existing Account

```
                SEARCH RECORD
Enter Account Number Which Details you Want to display: 1

                ACCOUNT DETAILS



-----------------------------------------------


Account No: 1
Name: Kaushal
Phone No: 769233805
Address: jalgaon
Created on (mm dd yyyy): 1 1 2021
DOB (mm dd yyyy): 22 4 2003
Bank Balance: 550000.000000
Account Type: #Saving
-----------------------------------------------
```

5. Manage Transaction

```
                MANAGE TRANSACTION


AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1               Kaushal     550000.000000       #Saving
2               Abhishek    50000.000000        #Current
3               Kaustubh    70000.000000        #Saving
4               Shantanu    45000.000000        #Current
5               Pavan       45000.000000        #Fixed1
6               Anurag      51000.000000        #Fixed2
7               Gaurav      78000.000000        #Saving
8               Shubham     45100.000000        #Current
9               Amarbir     90000.000000        #Fixed3
10              Bhupesh     41000.000000        #Current
-------------------------------------------------------
Enter Account No. For transaction: 1

        1.Withdrawal
        2.Deposite

        Enter your choice: 1

        Total Balance in 1 Account before Withdrawal = 550000.000000
        Enter Amount to withdrawal: 20000

        Total Balance in 1 Account after Withdrawal = 530000.000000
```

```
                MANAGE TRANSACTION


AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1               Kaushal     530000.000000       #Saving
2               Abhishek    50000.000000        #Current
3               Kaustubh    70000.000000        #Saving
4               Shantanu    45000.000000        #Current
5               Pavan       45000.000000        #Fixed1
6               Anurag      51000.000000        #Fixed2
7               Gaurav      78000.000000        #Saving
8               Shubham     45100.000000        #Current
9               Amarbir     90000.000000        #Fixed3
10              Bhupesh     41000.000000        #Current
-------------------------------------------------------
Enter Account No. For transaction: 10

        1.Withdrawal
        2.Deposite

        Enter your choice: 2
        Total Balance in 10 Account before Deposite = 41000.000000
        Enter Amount to Deposite: 900

Total Balance in 10 Account after Deposite = 41900.000000
```

6. Delete Account From Bank

```
                DELETE RECORD

AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1               Kaushal     530000.000000       #Saving
2               Abhishek    50000.000000        #Current
3               Kaustubh    70000.000000        #Saving
4               Shantanu    45000.000000        #Current
5               Pavan       45000.000000        #Fixed1
6               Anurag      51000.000000        #Fixed2
7               Gaurav      78000.000000        #Saving
8               Shubham     45100.000000        #Current
9               Amarbir     90000.000000        #Fixed3
10              Bhupesh     41900.000000        #Current
-------------------------------------------------------

        Enter Account number to delete: 6

        Anurag Account Removed From Bank!!!
```

7. Display All Record

```
AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1               Kaushal     530000.000000       #Saving
2               Abhishek    50000.000000        #Current
3               Kaustubh    70000.000000        #Saving
4               Shantanu    45000.000000        #Current
5               Pavan       45000.000000        #Fixed1
6               Anurag      51000.000000        #Fixed2
7               Gaurav      78000.000000        #Saving
8               Shubham     45100.000000        #Current
9               Amarbir     90000.000000        #Fixed3
10              Bhupesh     41900.000000        #Current
-------------------------------------------------------
```

Conclusion

In this way we have learn about bank management system and its benefits. We have implement the program using linked list.

linked list is a linear data structure. We used different header files and function available in those header file to implement program.

we practice different operation on linked list such as insertion, deletion, traversing and searching using c language.

Future Scope:

1. Using File Handling
2. By Using Concept of Object Oriented Programming reliability and reusability increase.
3. We can use user level and administrator level authentication.

References

1. https://www.geeksforgeeks.org/time-h-header-file-in-c-with-examples - Use for adding time delay to the project
2. http://www.cplusplus.com/forum/windows/166491 - use for formatted the output
3. https://www.sanfoundry.com/c-program-illustrate-user-authentication - use for authenticated code
4. Data Structure & Algorithum using c - use for dynamic memory allocation
5. https://www.upgrad.com/blog/data-structure-project-ideas-beginners - use for project idea

Course Outcomes Achieved

1. Implement basic operation on linked list.
2. Write an algorithm to Search the given node using linear Search.
3. Create relevant Structure to represent node using linked list


HAVE ANY
QUERY

Thank you