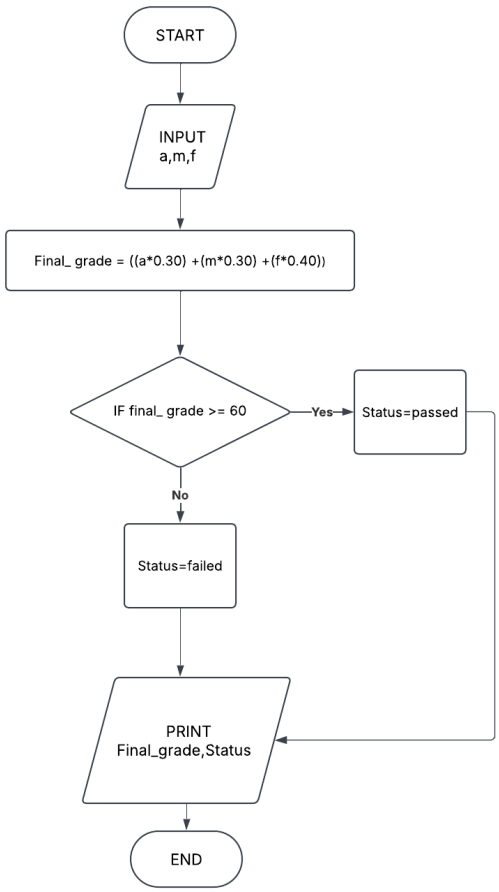
1. Student Grade Calculator

Algorithm:
Start
Input assignment, midterm and final marks
Find weighted average by adding assignment mark, Midterm mark and final score after multiplying them with 0.30,0.30 and 0.40 respectively.
If weighted average is greater than or equal to 60 set status to pass, otherwise set status to Fail
Display final grade and status
End

Pseudocode:

```
START
        PRINT enter assignment, midterm and final marks
        INPUT a, m, f
        Final_ grade = ((a*0.30) +(m*0.30) +(f*0.40))
        IF final_ grade >= 60
                Status=passed
        ELSE
                Status=failed
        END IF
        PRINT Final_ grade, Status
END
```

Flowchart:



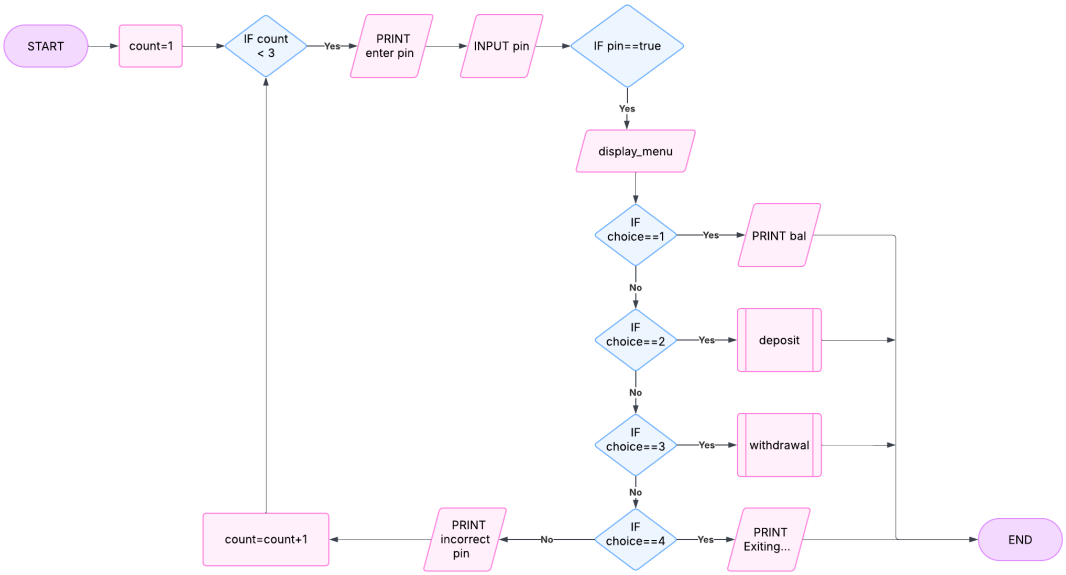
2. ATM Banking System

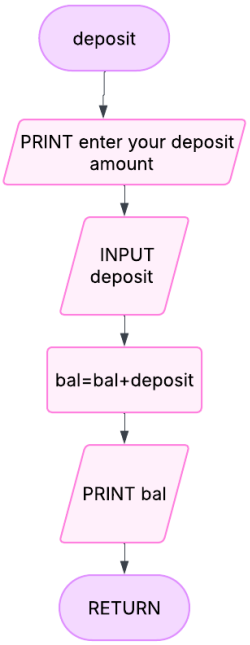
Algorithm:
Start
Set initial balance
Authenticate user with PIN
Display menu:
Check balance
Deposit money
Withdraw money
Exit
If deposit, add amount to balance
If withdraw, check sufficient funds:
If yes, deduct from balance
If no, print "Insufficient funds"
Loop until user chooses to exit
End

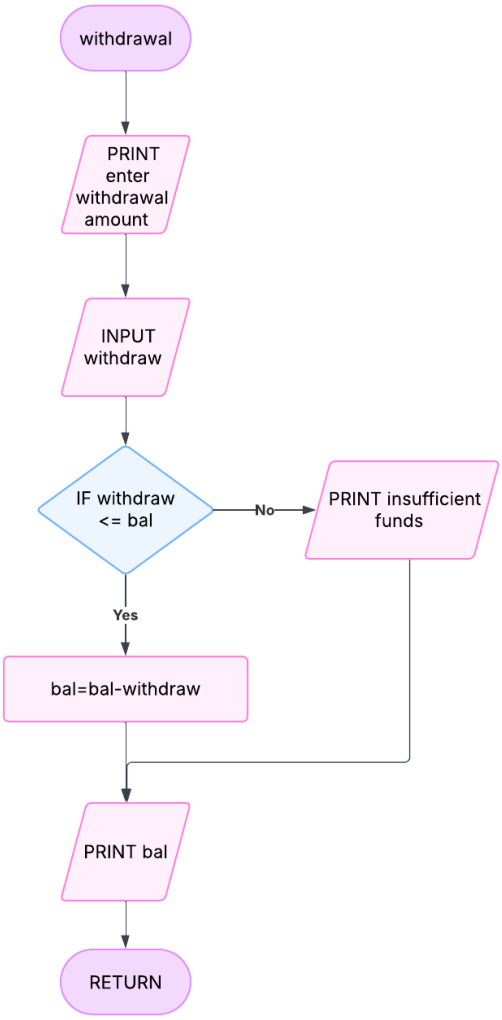
Pseudocode

```
START
   SET balance = 1000
   PRINT "Enter PIN"
   INPUT pin
   IF pin is correct THEN
      DO
         PRINT "1. Check Balance, 2. Deposit, 3. Withdraw, 4. Exit"
         INPUT choice
         CASE choice OF
            "1": PRINT "Balance: ", balance
            "2": PRINT "Enter deposit amount"
                 INPUT deposit
                 balance = balance + deposit
                 PRINT "New Balance: ", balance
            "3": PRINT "Enter withdrawal amount"
                 INPUT withdraw
                 IF withdraw <= balance THEN
                   balance = balance - withdraw
                   PRINT "New Balance: ", balance
                 ELSE
                   PRINT "Insufficient Funds"
                 ENDIF
            "4": PRINT "Exiting..."
                 EXIT
           DEFAULT: PRINT "Invalid Option"
         END CASE
      WHILE choice != 4
   ELSE
      PRINT "Incorrect PIN"
   ENDIF
END
```

Flowchart:


START
count=1
IF count < 3
Yes
PRINT enter pin
INPUT pin
IF pin==true
Yes
display_menu
IF choice==1
Yes
PRINT bal
No
IF choice==2
Yes
deposit
No
IF choice==3
Yes
withdrawal
No
IF choice==4
Yes
PRINT Exiting...
END
No
PRINT incorrect pin
count=count+1


deposit
PRINT enter your deposit amount
INPUT deposit
bal=bal+deposit
PRINT bal
RETURN


withdrawal
PRINT enter withdrawal amount
INPUT withdraw
IF withdraw <= bal
No
PRINT insufficient funds
Yes
bal=bal-withdraw
PRINT bal
RETURN

3. Inventory Management System

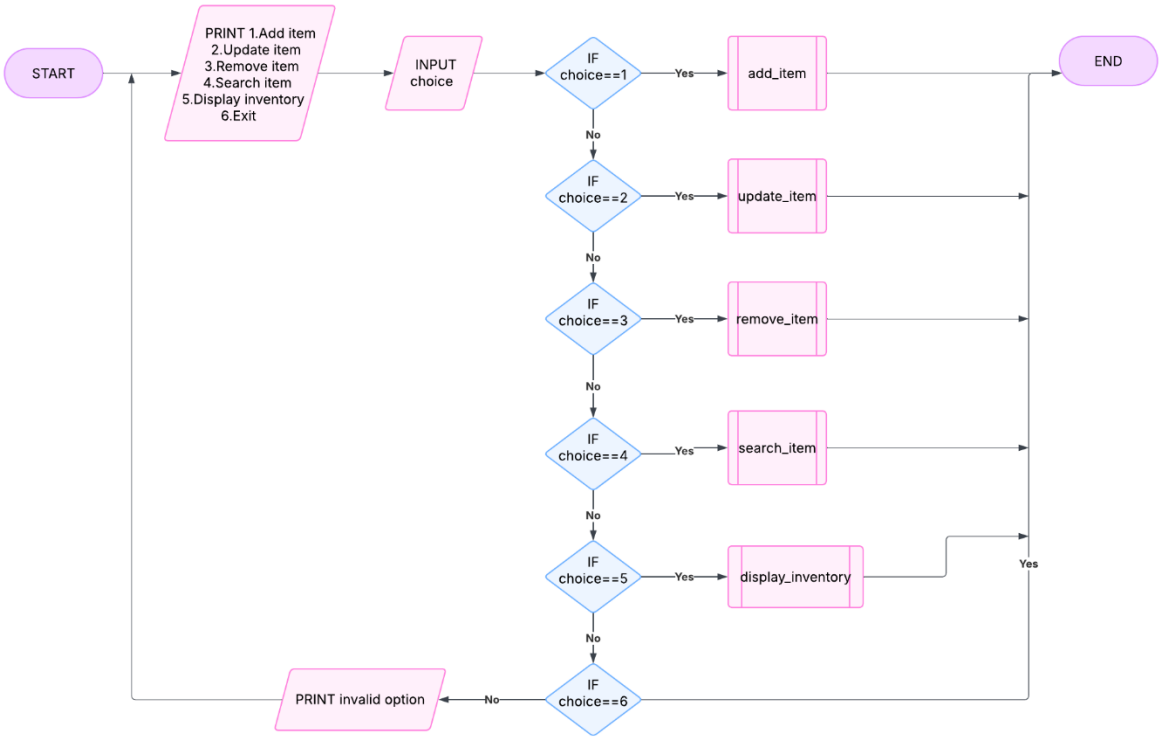
Algorithm:

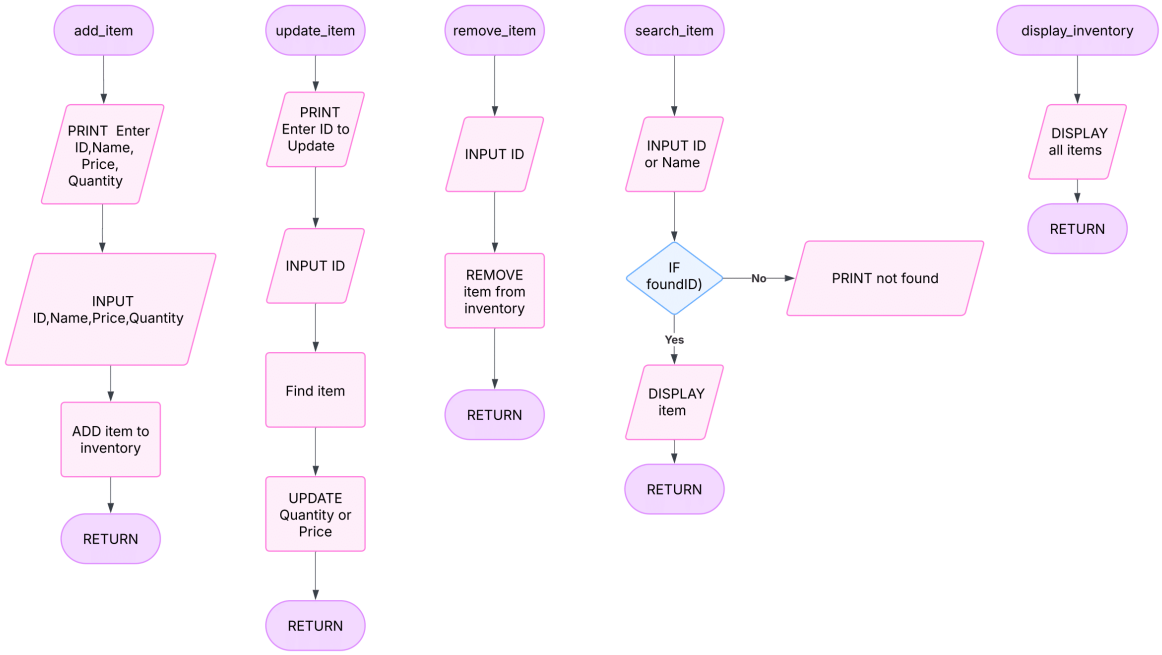
Start
Initialize inventory list
Display menu:
Add item
Update item
Remove item
Search item
Display inventory
Exit
Perform operations based on user input
Loop until exit
End

Pseudocode:

```
START
  DEFINE inventory as empty list
  DO
    PRINT "1. Add Item, 2. Update Item, 3. Remove Item, 4. Search Item, 5. Display Inventory, 6. Exit"
    INPUT choice
    CASE choice OF
      "1": PRINT "Enter ID, Name, Price, Quantity"
        INPUT id, name, price, quantity
        ADD item to inventory
      "2": PRINT "Enter ID to update"
        INPUT id
        FIND item
        UPDATE quantity or price
      "3": PRINT "Enter ID to remove"
        INPUT id
        REMOVE item from inventory
      "4": PRINT "Enter ID or Name to search"
        INPUT search
        DISPLAY item if found
      "5": DISPLAY all items
      "6": PRINT "Exiting..."
        EXIT
      DEFAULT: PRINT "Invalid Option"
    END CASE
  WHILE choice != 6
END
```

Flowchart:


START
PRINT 1.Add item
2.Update item
3.Remove item
4.Search item
5.Display inventory
6.Exit
INPUT choice
IF choice==1
Yes
add_item
END
No
IF choice==2
Yes
update_item
No
IF choice==3
Yes
remove_item
No
IF choice==4
Yes
search_item
No
IF choice==5
Yes
display_inventory
Yes
No
IF choice==6
No
PRINT invalid option


add_item
PRINT Enter ID,Name, Price, Quantity
INPUT ID,Name,Price,Quantity
ADD item to inventory
RETURN
update_item
PRINT Enter ID to Update
INPUT ID
Find item
UPDATE Quantity or Price
RETURN
remove_item
INPUT ID
REMOVE item from inventory
RETURN
search_item
INPUT ID or Name
IF foundID)
No
PRINT not found
Yes
DISPLAY item
RETURN
display_inventory
DISPLAY all items
RETURN

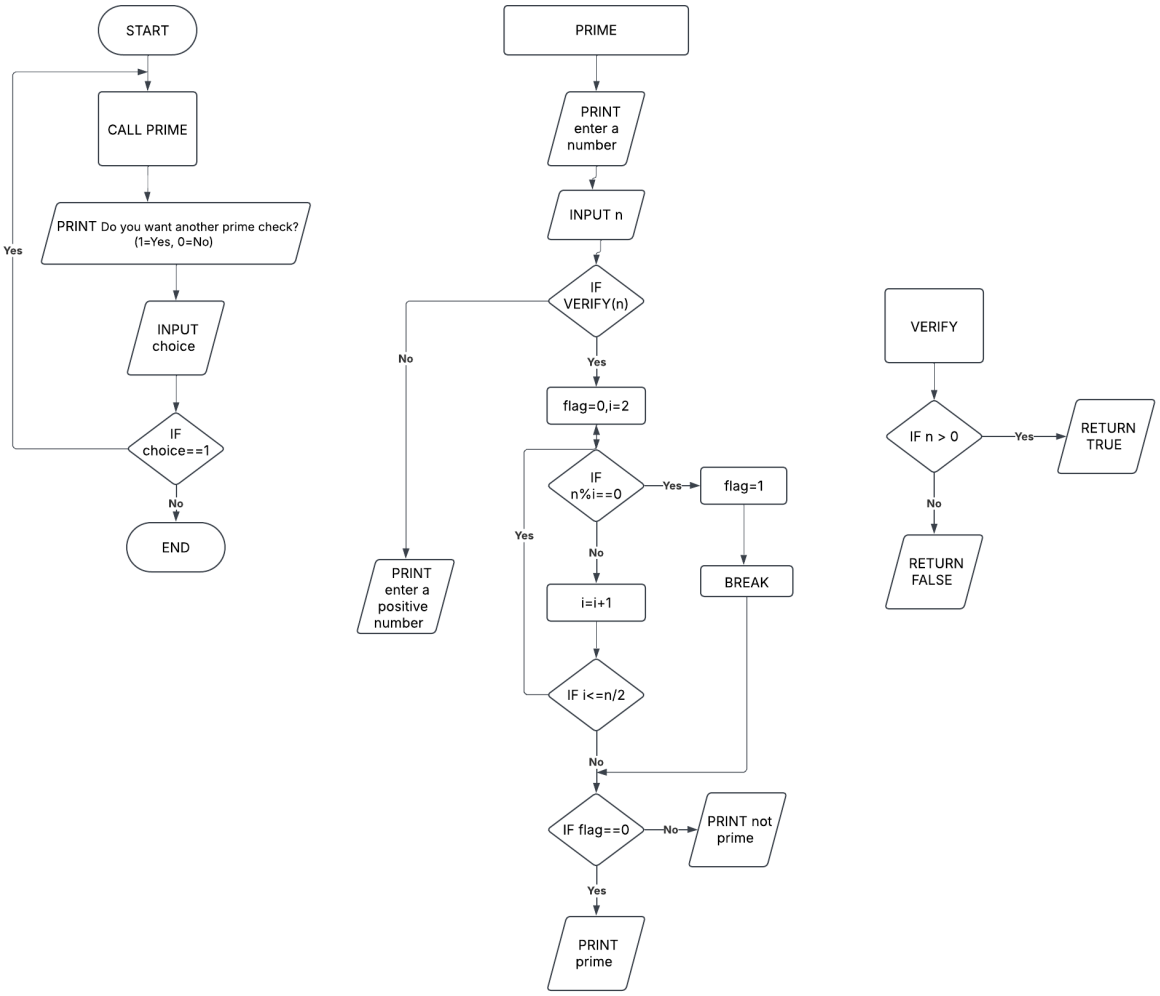
4. Prime Number Checker

Algorithm:
Start
Input a number
Check whether the number is positive or not, if not request for reinput of number
If the number is completely divisible by any other number except 1 and the number itself then set flag to 1
If flag is 0 then the number is prime, otherwise it is not a prime number
Check whether an intent for another prime check and repeat the whole prime check
End

Pseudocode:

```
START
   CALL PRIME
   PRINT Do you want another prime check? (1=Yes, 0=No)
   INPUT choice
   IF choice == 1 THEN
      CALL PRIME
   ELSE
      END
END

PRIME
   PRINT Enter a number
   INPUT n
   IF VERIFY(n) THEN
      flag = 0
      FOR i  = 2 TO n/2
         IF n % i == 0 THEN
            flag = 1
            BREAK
         END IF
      END FOR
      IF flag == 0 THEN
         PRINT n is a prime number
      ELSE
         PRINT n is not a prime number
      END IF
   ELSE
      PRINT Please enter a positive integer.
      CALL PRIME
END PRIME

VERIFY(n)
   IF n > 0 THEN
      RETURN TRUE
   ELSE
      RETURN FALSE
END VERIFY
```

Flowchart:



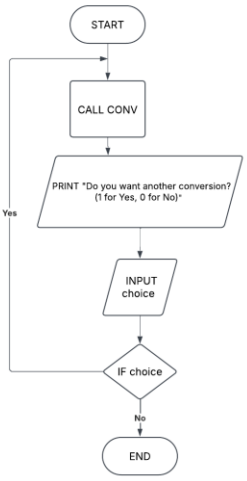
5. Temperature Conversion Tool

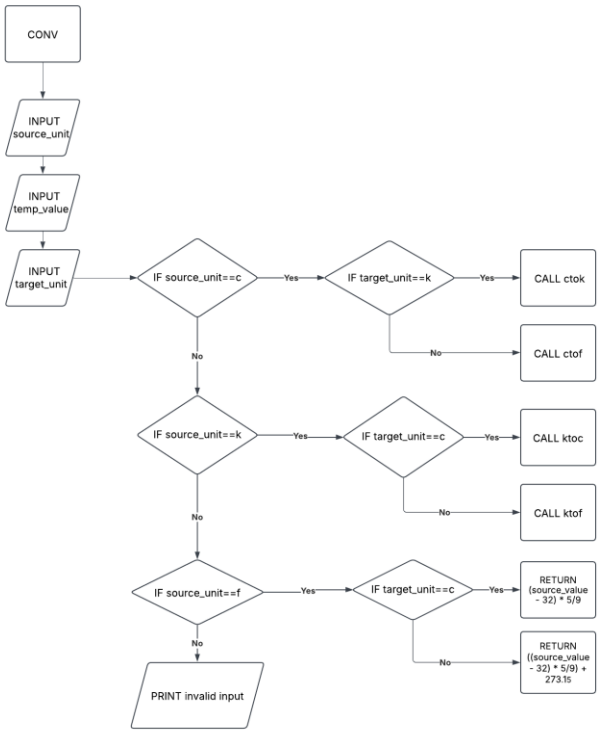
Algorithm:
Start
Input source_unit and target_unit
Input temp_value
According to source and target unit case executed accordingly
Check whether an intent for another prime check and repeat the whole prime check
End

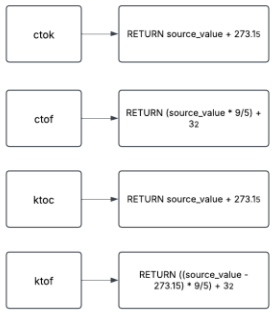
Pseudocode:

```
START
   CALL CONV
   PRINT "Do you want another conversion? (1 for Yes, 0 for No)"
   INPUT choice
   IF choice = 1
      CALL CONV
   END IF
END

CONV()
   PRINT Select Source Unit
   INPUT source_unit
   PRINT "Enter Temperature Value"
   INPUT source_value
   PRINT Select Target Unit
   INPUT target_unit

   CASE (source_unit, target_unit)
      CASE ("c", "k"): OUTPUT ctok(source_value)
      CASE ("c", "f"): OUTPUT ctof(source_value)
      CASE ("k", "c"): OUTPUT ktoc(source_value)
      CASE ("k", "f"): OUTPUT ktof(source_value)
      CASE ("f", "c"): OUTPUT ftoc(source_value)
      CASE ("f", "k"): OUTPUT ftok(source_value)
      DEFAULT:
         PRINT Invalid Input
   END CASE
END CONV

ctok(source_value)
   RETURN source_value + 273.15

ctof(source_value)
   RETURN (source_value * 9/5) + 32

ktoc(source_value)
   RETURN source_value - 273.15

ktof(source_value)
   RETURN ((source_value - 273.15) * 9/5) + 32

ftoc(source_value)
   RETURN (source_value - 32) * 5/9

ftok(source_value)
   RETURN ((source_value - 32) * 5/9) + 273.15
```

Flowchart:







6. Library Book Management System

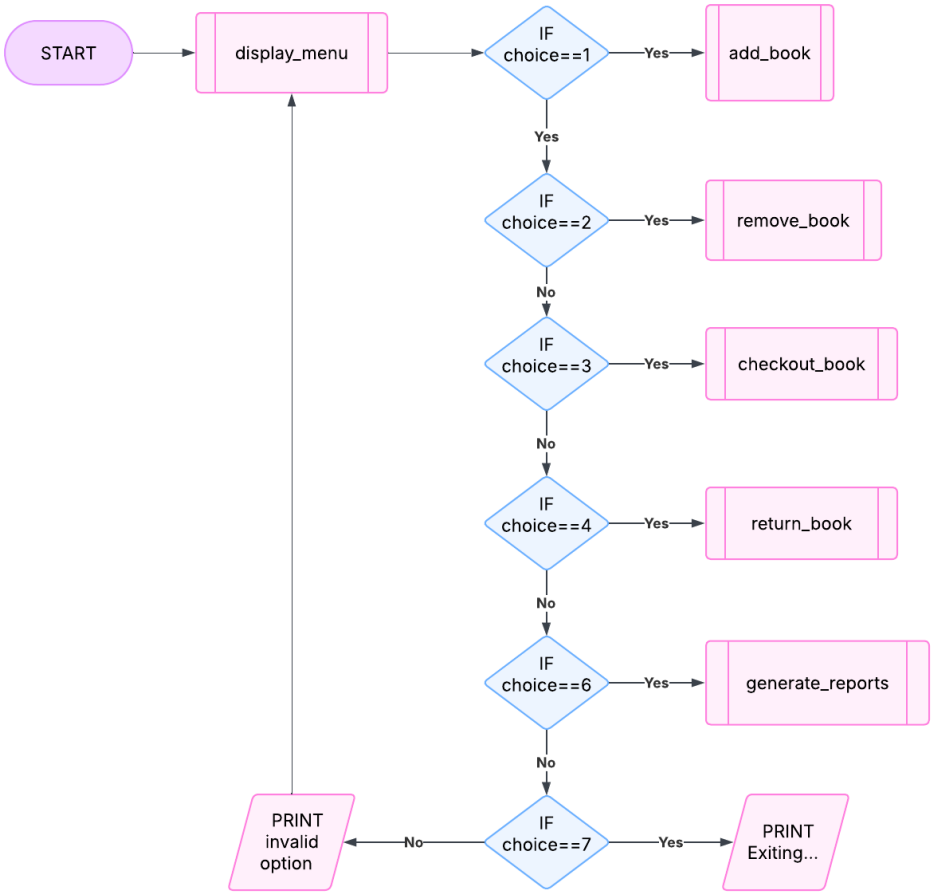
Algorithm:
Start
Initialize book and member databases
Display menu:
Add book
Remove book
Checkout book
Return book
Search book
Generate reports
Exit
Perform operations based on user input
Loop until exit
End

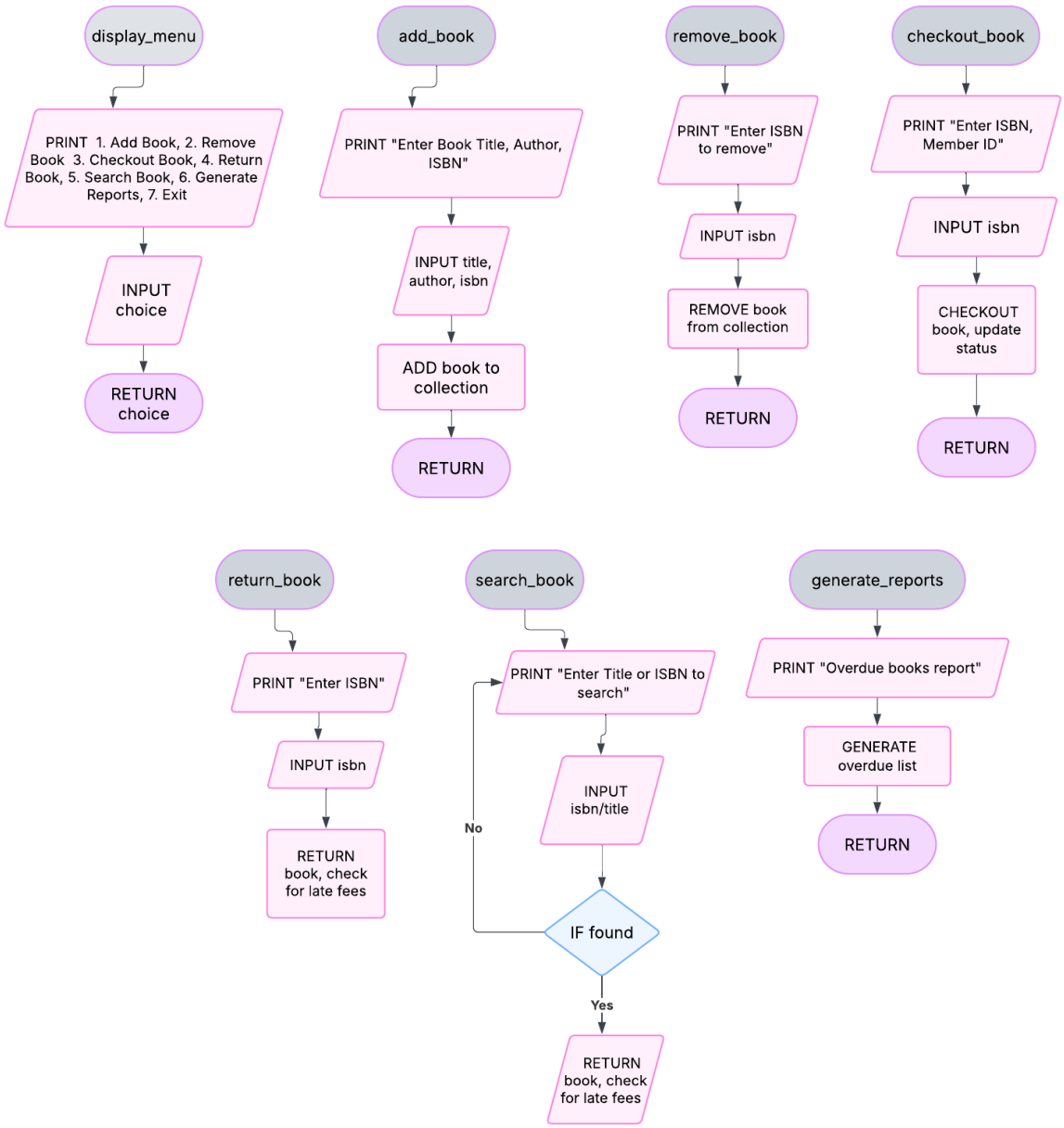
Pseudocode:

```
START
   DEFINE books, members as empty lists
   DO
      PRINT "1. Add Book, 2. Remove Book, 3. Checkout Book, 4. Return Book, 5. Search Book, 6. Generate Reports, 7. Exit"
      INPUT choice
      CASE choice OF
         "1": PRINT "Enter Book Title, Author, ISBN"
```

```
            INPUT title, author, isbn
            ADD book to collection
        "2": PRINT "Enter ISBN to remove"
            INPUT isbn
            REMOVE book from collection
        "3": PRINT "Enter ISBN, Member ID"
            INPUT isbn, member_id
            CHECKOUT book, update status
        "4": PRINT "Enter ISBN"
            INPUT isbn
            RETURN book, check for late fees
        "5": PRINT "Enter Title or ISBN to search"
            INPUT search
            DISPLAY book details if found
        "6": PRINT "Overdue books report"
            GENERATE overdue list
        "7": PRINT "Exiting..."
            EXIT
        DEFAULT: PRINT "Invalid Option"
    END CASE
  WHILE choice != 7
END
```

Flowchart:





7. Fibonacci Sequence Generator

Algorithm:

Start

Input number of terms

Check if input is valid

Initialize first two Fibonacci numbers

Loop to generate sequence

Print sequence

End

Pseudocode

```
START
   PRINT "Enter number of terms"
   INPUT n
   IF n < 0 THEN
```

```
    PRINT "Invalid input"
    EXIT
  ENDIF
  SET a = 0, b = 1
  PRINT a, b
  FOR i = 2 TO n
    SET c = a + b
    PRINT c
    SET a = b
    SET b = c
  END FOR
END
```

8. Calendar Event Scheduler

Algorithm:

Start

Initialize event database

Display menu:

Add event

View events

Delete event

Search event

Exit

Pseudocode:

```
START
  DEFINE events as empty list
  DO
    PRINT "1. Add Event, 2. View Events, 3. Delete Event, 4. Search Event, 5. Exit"
    INPUT choice
    CASE choice OF
      "1": PRINT "Enter Title, Date, Time, Description"
           INPUT title, date, time, description
           ADD event to list
      "2": PRINT "Enter Date to view events"
           INPUT date
           DISPLAY events on that date
      "3": PRINT "Enter Event Title to delete"
           INPUT title
           REMOVE event if found
      "4": PRINT "Enter keyword to search"
           INPUT keyword
           DISPLAY matching events
      "5": PRINT "Exiting..."
           EXIT
      DEFAULT: PRINT "Invalid Option"
    END CASE
  WHILE choice != 5
END
```

Flowchart: