→ "Library Management System" :→

```
Class Book:
    def __init__(self, book_id, title, author, copies):
        self.book_id = book_id
        self.title = title
        self.author = author
        self.copies = copies

    def display_info(self):
        print(f"ID: {self.book_id}, Title: {self.title}, Author: {self.author},
         Copies Available: {self.copies}")


Class Library:
    def __init__(self):
        self.books = [ ]

    def add_book(self, book_id, title, author, copies):
        new_book = Book(book_id, title, author, copies)
        self.books.append(new_book)
        print("Book added successfully.")

    def display_books(self):
        if not self.books:
            print("No books available in the library.")
        else:
            print("Books in Library:")
            for book in self.books:
                book.display_info()

    def search_book(self, title):
        found = False.
```

[P.T.O]

```
    for book in self.books:
        if book.title.lower() == title.lower():
            book.display_info()
            found = True
            break
    if not found:
        print("Book not found.")

def borrow_book(self, title):
    for book in self.books:
        if book.title.lower() == title.lower():
            if book.copies > 0:
                book.copies -=1
                print("Book borrowed successfully.")
            else:
                print("Book is not available.")
            return.
    print ("Book not found.")

def return_book(self, title):
    for book in self.books:
        if book.title.lower() == title.lower():
            book.copies +=1
            print("Book returned successfully.")
            return
    print("Book not found.").

library = Library()
while True:
```

```
print('\n === Library Management System ===")
print("1. Add Book")
print("2. Display All Books")
print("3. Search Book")
print("4. Borrow Book")
print("5. Return Book")
print("6. Exit")

Choice = input("Enter your choice:")

if choice == '1':
    book_id = input("Enter Book ID:")
    title = input("Enter Book Title:")
    author = input("Enter Book Author:")
    copies = int(input("Enter number of Copies:"))
    library.add_book(book_id, title, author, copies)

elif choice == '2':
    library.display_books()

elif choice == '3':
    title = input("Enter book title to search:")
    library.search_book(title)

elif choice == '4':
    title = input("Enter book title to borrow:")
    library.borrow_book(title)

elif choice == '5':
    title = input("Enter book title to return:")
    library.return_book(title)

elif choice == '6':
    print("Exiting Program. Good Bye!")
    break.
```

```
else:
    point("Invalid choice, Please try again.")
```

→ "Menu based Bank Management System".

```
Class Account:
  def __init__(self, account_number, name, balance):
      self.account_number = account_number
      self.name = name
      self.balance = balance

  def display_account(self):
      print(f"Account Number : {self.account_number}")
      print(f"Account Holder : {self.name}")
      print(f"Balance        : {self.balance}")


  def deposit(self, amount):
      if amount > 0:
         self.balance .+= amount
         print(f"Amount Deposited Successfully. New Balance = {self.balance}")
      else:
         print("Invalid Amount! Enter a positive Value.")

  def withdraw(self, amount):
      if amount > 0:
         If self.balance >= amount:
            self.balance -= amount
            print(f"Amount withdraw Successfully. Remaining Balance =
               {self.balance}")
```

```
        else:
            print("Insufficient Balance!")
    else:
        print ("Invalid Amount! Enter a positive value.")

    def check_balance(self):
        print (f"Available Balance = {self.balance}")

class Bank:
    def __init__(self):
        self.accounts = []

    def create_account(self, account_number, name, balance):
        new_account = Account(account_number, name, balance)
        self.accounts.append (new_account)
        print("Account created successfully.")

    def find_account(self, account_number):
        for account in self.accounts:
            if account.account_number == account_number:
                return account.

        return None

bank = Bank()

While True:
    print("\n ==== Bank Management System ====")
    print("1. Create Account")
    print ("2. View Account Details")
    print("3. Deposit Money")
    print ("4. Withdraw Money")
```

```
print("5. Check Balance")
print("6. Exit")
Choice = input('Enter your choice: ")
if choice == '1':
    acc_no = input("Enter Account Number:")
    name = input("Enter Account Holder Name: ")
    balance = float(input("Enter Initial Balance: "))
    bank.create_account(acc_no, name, balance)

elif choice == '2':
    acc_no = input("Enter Account Number:")
    account = bank.find_account(acc_no)
    if account:
        account.display_account()
    else:
        print("Account not found!")

elif choice == '3':
    acc_no = input("Enter Account Number: ")
    account = bank.find_account(acc_no)
    if account:
        amount = float(input("Enter Amount to Deposit: "))
        account.deposit(amount)
    else:
        print("Account not found!")

elif choice == '4':
    acc_no = input("Enter Account Number: ")
    account = bank.find_account (acc_no).
    if account:
```

P.T.O

```
amount = float(input("Enter amount to withdraw: "))
account. withdraw(amount)
else:
    point("Account Not found!")

elif choice == '5':
    acc_no = input("Enter Account Number: ")
    account = banp. find_account(acc_no)
    if account:
        account. checp_balance()
    else:
        point("Account Not Found!")

elif choice == '6':
    print("Exiting Program. Thanp You!")
    breap

else:
    point("Invalid choice! Please Try again.")
```

———o———