Question 1 :- Student Database with Percentage and Grade

Problem Statement:

Create a student database using lists and dictionaries. Each student has a name, roll number, and marks in three subjects. Implement the following operations:

1. Add a new student to the database.
2. Display the details of a specific student.
3. Display the average marks of all students.
4. Display the percentage of all students.
5. Display the grade of a specific student.

Sample Input:

1. Add new student
2. Display student details
3. Display average marks
4. Display percentage of all students
5. Display grade of a specific student
6. Exit

Enter your choice: 1

Enter student name: Alice

Enter roll number: 101

Enter marks in Subject 1: 85

Enter marks in Subject 2: 90

Enter marks in Subject 3: 92

Enter your choice: 1

Enter student name: Bob

Enter roll number: 102

Enter marks in Subject 1: 78

Enter marks in Subject 2: 88

Enter marks in Subject 3: 95

Enter your choice: 3

Sample Output:

Average Marks:

Subject 1: 81.5

Subject 2: 89.0

Subject 3: 93.5

Enter your choice: 4

Percentage of Students:

Alice (Roll Number 101): 89.0%

Bob (Roll Number 102): 87.0%

Enter your choice: 5

Enter roll number of the student: 101

Grade of Student:

Name: Alice

Roll Number: 101

Percentage: 89.0%

Grade: A

Solution :-

```
# Initialize an empty list to store student records
student_database = []

# Choice-based menu
while True:
    print("\nStudent Database Menu:")
    print("1. Add new student")
    print("2. Display student details")
    print("3. Display average marks")
    print("4. Display percentage of all students")
    print("5. Display grade of a specific student")
    print("6. Exit")

    choice = input("Enter your choice (1-6): ")

    if choice == '1':
        # Add new student
        student = {}
        student["Name"] = input("Enter student name: ")
        student["Roll Number"] = input("Enter roll number: ")
        student["Marks"] = [int(input(f"Enter marks in Subject {i + 1}: ")) for i in range(3)]

        student_database.append(student)
        print(f"Student {student['Name']} added successfully.")

    elif choice == '2':
        # Display student details
        roll_number = input("Enter roll number of the student: ")
        found = False
```

```
        for student in student_database:
            if student["Roll Number"] == roll_number:
                print("Student Details:")
                print(f"Name: {student['Name']}")
                print(f"Roll Number: {student['Roll Number']}")
                print(f"Marks: {student['Marks']}")
                found = True
                break

        if not found:
            print(f"No student found with Roll Number: {roll_number}")

    elif choice == '3':
        # Display average marks
        averages = [sum(student["Marks"][i] for student in student_database) /
len(student_database) for i in range(3)]
        print("Average Marks:")
        for i, average in enumerate(averages):
            print(f"Subject {i + 1}: {average}")

    elif choice == '4':
        # Display percentage of all students
        print("Percentage of Students:")
        for student in student_database:
            percentage = sum(student["Marks"]) / len(student["Marks"])
            print(f"{student['Name']} (Roll Number {student['Roll Number']}): {percentage}%")

    elif choice == '5':
        # Display grade of a specific student
        roll_number = input("Enter roll number of the student: ")
```

```
        found = False

        for student in student_database:
            if student["Roll Number"] == roll_number:
                percentage = sum(student["Marks"]) / len(student["Marks"])
                grade = 'A' if percentage >= 90 else 'B' if percentage >= 80 else 'C' if percentage >= 70 else 'D' if percentage >= 60 else 'F'
                print("Grade of Student:")
                print(f"Name: {student['Name']}")
                print(f"Roll Number: {student['Roll Number']}")
                print(f"Percentage: {percentage}%")
                print(f"Grade: {grade}")
                found = True
                break

        if not found:
            print(f"No student found with Roll Number: {roll_number}")

    elif choice == '6':
        # Exit the program
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 6.")
```

Question 2 :-

Problem Statement: Patient Database

You are tasked with creating an interactive patient database system in Python. The system should allow the user to choose from various options such as adding new patients, retrieving patient information, updating patient records, and displaying information for all patients. Each patient has the following attributes:

1. **Patient ID:** A unique identifier for each patient.
2. **Name:** The name of the patient.
3. **Age:** The age of the patient.
4. **Gender:** The gender of the patient.
5. **Contact Number:** The contact number of the patient.
6. **Medical History:** A list of medical conditions or history for the patient.

Requirements:

1. Implement a choice-based menu system that allows the user to select operations:
    - Option 1: Add new patient.
    - Option 2: Retrieve patient information based on Patient ID.
    - Option 3: Update the medical history of a patient based on Patient ID.
    - Option 4: Display information of all patients.
    - Option 5: Exit the program.
2. Use the user's input to perform the selected operation.

Sample Input:

```
# Initial patient database
patient_database = []


# Choice-based menu
……..
……

    choice = input("Enter your choice (1-5): ")
```

```
    if choice == '1':
        # User input to add a new patient

    elif choice == '2':
        # User input to retrieve patient information

    elif choice == '3':
        # User input to update patient medical history

    elif choice == '4':
        # Display information of all patients

    elif choice == '5':
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 5.")
```

Sample Output:

Patient Database Menu:

1. Add new patient
2. Retrieve patient information
3. Update patient medical history
4. Display all patients
5. Exit

Enter your choice (1-5): 1

User provides information to add a new patient

Patient added successfully.

Patient Database Menu:

1. Add new patient

2. Retrieve patient information

3. Update patient medical history

4. Display all patients

5. Exit

Enter your choice (1-5): 2

User provides information to retrieve patient information

Patient Database Menu:

1. Add new patient

2. Retrieve patient information

3. Update patient medical history

4. Display all patients

5. Exit

Enter your choice (1-5): 3

User provides information to update patient medical history

Medical history updated for patient ID [ID].

Patient Database Menu:

1. Add new patient

2. Retrieve patient information

3. Update patient medical history

4. Display all patients

5. Exit

Enter your choice (1-5): 4

Display information of all patients

Patient Database Menu:

1. Add new patient

2. Retrieve patient information

3. Update patient medical history

4. Display all patients

5. Exit

Enter your choice (1-5): 5

Exiting the program.

Solution :-

```
# Initial patient database
patient_database = []

# Choice-based menu
while True:
    print("\nPatient Database Menu:")
    print("1. Add new patient")
    print("2. Retrieve patient information")
    print("3. Update patient medical history")
    print("4. Display all patients")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == '1':
        # User input to add a new patient
        patient_id = input("Enter Patient ID: ")
        name = input("Enter Name: ")
        age = input("Enter Age: ")
        gender = input("Enter Gender: ")
        contact_number = input("Enter Contact Number: ")
        medical_history = input("Enter Medical History (comma-separated): ").split(',')
```

```
        patient = {
            "Patient ID": patient_id,
            "Name": name,
            "Age": age,
            "Gender": gender,
            "Contact Number": contact_number,
            "Medical History": medical_history
        }

        patient_database.append(patient)
        print("Patient added successfully.")

    elif choice == '2':
        # User input to retrieve patient information
        patient_id = input("Enter Patient ID: ")
        found = False

        for patient in patient_database:
            if patient["Patient ID"] == patient_id:
                print("Patient Information:")
                for key, value in patient.items():
                    print(f"{key}: {value}")
                found = True
                break

        if not found:
            print(f"No patient found with Patient ID: {patient_id}")

    elif choice == '3':
        # User input to update patient medical history
        patient_id = input("Enter Patient ID: ")
```

```
        found = False

        for patient in patient_database:
            if patient["Patient ID"] == patient_id:
                new_medical_history = input("Enter Updated Medical History (comma-separated): ").split(',')
                patient["Medical History"] = new_medical_history
                print(f"Medical history updated for patient ID {patient_id}.")
                found = True
                break

        if not found:
            print(f"No patient found with Patient ID: {patient_id}")

    elif choice == '4':
        # Display information of all patients
        if not patient_database:
            print("No patients in the database.")
        else:
            print("All Patients:")
            for patient in patient_database:
                print(f"Patient ID: {patient['Patient ID']}, Name: {patient['Name']}, Age: {patient['Age']}, Gender: {patient['Gender']}, Contact Number: {patient['Contact Number']}, Medical History: {patient['Medical History']}")

    elif choice == '5':
        # Exit the program
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 5.")
```

Question 3 :-

Problem Statement: Employee Database

You are tasked with creating an interactive employee database system in Python. The system should allow the user to choose from various options such as adding new employees, retrieving employee information, updating employee records, and displaying information for all employees. The employee details include:

1. **Employee ID:** A unique identifier for each employee.
2. **Name:** The name of the employee.
3. **Age:** The age of the employee.
4. **Gender:** The gender of the employee.
5. **Position:** The job position or role of the employee.
6. **Salary:** The monthly salary of the employee.

Requirements:

1. Implement a choice-based menu system that allows the user to select operations:
    - Option 1: Add new employees.
    - Option 2: Retrieve employee information based on Employee ID.
    - Option 3: Update the salary of an employee based on Employee ID.
    - Option 4: Display information of all employees.
    - Option 5: Exit the program.
2. Use the user's input to perform the selected operation.

Sample Output:

Employee Database Menu:

1. Add new employee

2. Retrieve employee information

3. Update employee salary

4. Display all employees

5. Exit

Enter your choice (1-5): 1

Enter Employee ID: 101

Enter the name of the employee: John Doe

Enter the age of the employee: 30

Enter the gender of the employee: Male

Enter the position of the employee: Software Engineer

Enter the salary of the employee: 70000

Employee added successfully.

Employee Database Menu:

1. Add new employee

2. Retrieve employee information

3. Update employee salary

4. Display all employees

5. Exit

Enter your choice (1-5): 2

Enter Employee ID to retrieve information: 101

Employee Information for ID 101:

Name: John Doe

Age: 30

Gender: Male

Position: Software Engineer

Salary: 70000

Employee Database Menu:

1. Add new employee

2. Retrieve employee information

3. Update employee salary

4. Display all employees

5. Exit

Enter your choice (1-5): 3

Enter Employee ID to update salary: 101

Enter the new salary: 75000

Salary updated for Employee ID 101.

Employee Database Menu:

1. Add new employee
2. Retrieve employee information
3. Update employee salary
4. Display all employees
5. Exit

Enter your choice (1-5): 4

All Employees:

Employee ID: 101

Name: John Doe

Age: 30

Gender: Male

Position: Software Engineer

Salary: 75000

Employee Database Menu:

1. Add new employee
2. Retrieve employee information
3. Update employee salary
4. Display all employees
5. Exit

Enter your choice (1-5): 5

Exiting the program.

Solution :-

```
# Initialize an empty list to store employee records
employee_database = []

# Choice-based menu
while True:
    print("\nEmployee Database Menu:")
    print("1. Add new employee")
    print("2. Retrieve employee information")
    print("3. Update employee salary")
    print("4. Display all employees")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == '1':
        # Add new employee
        employee_id = input("Enter Employee ID: ")
        name = input("Enter Name: ")
        age = input("Enter Age: ")
        position = input("Enter Position: ")
        salary = input("Enter Salary: ")

        employee = {
            "Employee ID": employee_id,
            "Name": name,
            "Age": age,
            "Position": position,
            "Salary": salary
        }
```

```
        employee_database.append(employee)
        print(f"Employee added successfully. Employee ID: {employee_id}")

    elif choice == '2':
        # Retrieve employee information
        employee_id = input("Enter Employee ID: ")
        found = False

        for employee in employee_database:
            if employee["Employee ID"] == employee_id:
                print("Employee Information:")
                print(f"Employee ID: {employee['Employee ID']}")
                print(f"Name: {employee['Name']}")
                print(f"Age: {employee['Age']}")
                print(f"Position: {employee['Position']}")
                print(f"Salary: {employee['Salary']}")
                found = True
                break

        if not found:
            print(f"No employee found with Employee ID: {employee_id}")

    elif choice == '3':
        # Update employee salary
        employee_id = input("Enter Employee ID: ")
        found = False

        for employee in employee_database:
            if employee["Employee ID"] == employee_id:
                print(f"Updating salary for Employee ID: {employee_id}")
                employee["Salary"] = input("Enter new Salary: ")
```

```
            print("Employee salary updated.")
            found = True
            break

        if not found:
            print(f"No employee found with Employee ID: {employee_id}")

    elif choice == '4':
        # Display all employees
        if not employee_database:
            print("No employees in the database.")
        else:
            print("All Employees:")
            for employee in employee_database:
                print(f"Employee ID: {employee['Employee ID']}, Name: {employee['Name']}, Age: {employee['Age']}, Position: {employee['Position']}, Salary: {employee['Salary']}")

    elif choice == '5':
        # Exit the program
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 5.")
```

Question 4 :-

Problem Statement: Library Database

You are tasked with creating an interactive library database system in Python. The system should allow the user to choose from various options such as adding new books, retrieving book information, updating the number of available copies, and displaying information for all books. Each book has the following attributes:

1. **Book ID:** A unique identifier for each book.
2. **Title:** The title of the book.
3. **Author:** The author of the book.
4. **Genre:** The genre or category of the book.
5. **Available Copies:** The number of copies available in the library.

Requirements:

1. Implement a choice-based menu system that allows the user to select operations:
    - Option 1: Add new book.
    - Option 2: Retrieve book information based on Book ID.
    - Option 3: Update available copies for a book based on Book ID.
    - Option 4: Display information of all books.
    - Option 5: Exit the program.
2. Use the user's input to perform the selected operation.

Sample Input:

```
# Initial library database

library_database = []


# Choice-based menu

……..

……..


  if choice == '1':

     # User input to add a new book


  elif choice == '2':

     # User input to retrieve book information
```

```
    elif choice == '3':
        # User input to update available copies

    elif choice == '4':
        # Display information of all books

    elif choice == '5':
        print("Exiting the program.")
        break

    else:
        print("Invalid choice. Please enter a number between 1 and 5.")
```

Sample Output:

Library Database Menu:

1. Add new book
2. Retrieve book information
3. Update available copies
4. Display all books
5. Exit

Enter your choice (1-5): 1

User provides information to add a new book

Book added successfully. Book ID: 101

Library Database Menu:

1. Add new book
2. Retrieve book information
3. Update available copies
4. Display all books

5. Exit

Enter your choice (1-5): 2

User provides information to retrieve book information

Book Information for ID 101:

Title: The Great Gatsby

Author: F. Scott Fitzgerald

Genre: Fiction

Available Copies: 10

Library Database Menu:

1. Add new book

2. Retrieve book information

3. Update available copies

4. Display all books

5. Exit

Enter your choice (1-5): 3

User provides information to update available copies

Available copies updated for Book ID 101.

Library Database Menu:

1. Add new book

2. Retrieve book information

3. Update available copies

4. Display all books

5. Exit

Enter your choice (1-5): 4

Display information of all books

All Books:

Book ID: 101

Title: The Great Gatsby

Author: F. Scott Fitzgerald

Genre: Fiction

Available Copies: 9

Library Database Menu:

1. Add new book

2. Retrieve book information

3. Update available copies

4. Display all books

5. Exit

Enter your choice (1-5): 5

Exiting the program.

Solution :-

```
# Initial library database
library_database = []

# Choice-based menu
while True:
    print("\nLibrary Database Menu:")
    print("1. Add new book")
    print("2. Retrieve book information")
    print("3. Update available copies")
    print("4. Display all books")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == '1':
        # User input to add a new book
        book_id = input("Enter Book ID: ")
        title = input("Enter Title: ")
```

```
    author = input("Enter Author: ")
    genre = input("Enter Genre: ")
    available_copies = int(input("Enter Available Copies: "))

    book = {
        "Book ID": book_id,
        "Title": title,
        "Author": author,
        "Genre": genre,
        "Available Copies": available_copies
    }

    library_database.append(book)
    print(f"Book added successfully. Book ID: {book_id}")

elif choice == '2':
    # User input to retrieve book information
    book_id = input("Enter Book ID: ")
    found = False

    for book in library_database:
        if book["Book ID"] == book_id:
            print("Book Information:")
            for key, value in book.items():
                print(f"{key}: {value}")
            found = True
            break

    if not found:
        print(f"No book found with Book ID: {book_id}")
```

```
elif choice == '3':
    # User input to update available copies
    book_id = input("Enter Book ID: ")
    found = False

    for book in library_database:
        if book["Book ID"] == book_id:
            new_available_copies = int(input("Enter Updated Available Copies: "))
            book["Available Copies"] = new_available_copies
            print(f"Available copies updated for Book ID {book_id}.")
            found = True
            break

    if not found:
        print(f"No book found with Book ID: {book_id}")

elif choice == '4':
    # Display information of all books
    if not library_database:
        print("No books in the library.")
    else:
        print("All Books:")
        for book in library_database:
            for key, value in book.items():
                print(f"{key}: {value}")
            print()

elif choice == '5':
    # Exit the program
    print("Exiting the program.")
    break
```

```
else:
    print("Invalid choice. Please enter a number between 1 and 5.")
```