Assignment 3

3.1

Aim: Write a program to create a basic calculator with functions for addition, subtraction, multiplication, and division.

Code:

```
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        return "Error: Division by zero"
    return a / b

def main():
    while True:
        print("\nBasic Calculator")
        print("1. Add")
        print("2. Subtract")
        print("3. Multiply")
        print("4. Divide")
        print("5. Exit")
        choice = input("Choose an operation (1-5): ")

        if choice == "5":
            print("Exiting the calculator. Goodbye!")
            break

        num1 = float(input("Enter the first number: "))
        num2 = float(input("Enter the second number: "))

        if choice == "1":
            print("Result:", add(num1, num2))
        elif choice == "2":
            print("Result:", subtract(num1, num2))
        elif choice == "3":
            print("Result:", multiply(num1, num2))
        elif choice == "4":
```

```
            print("Result:", divide(num1, num2))
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

~/VSCode/Python | main ?3
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-1.py"

Basic Calculator
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Choose an operation (1-5): 1
Enter the first number: 12
Enter the second number: 23
Result: 35.0

Basic Calculator
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Choose an operation (1-5): 2
Enter the first number: 54
Enter the second number: 45
Result: 9.0

Basic Calculator
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Choose an operation (1-5): 3
Enter the first number: 12
Enter the second number: 2
Result: 24.0

Basic Calculator
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Choose an operation (1-5): 4
Enter the first number: 6
Enter the second number: 2
Result: 3.0

Basic Calculator
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Choose an operation (1-5): 5
Exiting the calculator. Goodbye!

~/VSCode/Python | main ?3
```

3.2

Aim: Write a recursive function to calculate the factorial of a number.

Code:

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

def main():
    while True:
        print("\nFactorial Calculator")
        print("1. Calculate Factorial")
        print("2. Exit")
        choice = input("Choose an option (1-2): ")

        if choice == "2":
            print("Exiting the factorial calculator. Goodbye!")
            break

        number = int(input("Enter a number to calculate its factorial:
"))
        print("Factorial of", number, "is:", factorial(number))

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-2.py"
 ~/VSCode/Python | main ?4
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-2.py"

Factorial Calculator
1. Calculate Factorial
2. Exit
Choose an option (1-2): 1
Enter a number to calculate its factorial: 3
Factorial of 3 is: 6

Factorial Calculator
1. Calculate Factorial
2. Exit
Choose an option (1-2): 2
Exiting the factorial calculator. Goodbye!

 ~/VSCode/Python | main ?4
```

3.3

Aim: Write functions to perform various list operations such as finding the maximum, minimum, sum, and average of a list of numbers.

Code:

```python
def find_max(lst):
    max_val = lst[0]
    for num in lst:
        if num > max_val:
            max_val = num
    return max_val

def find_min(lst):
    min_val = lst[0]
    for num in lst:
        if num < min_val:
            min_val = num
    return min_val

def find_sum(lst):
    total = 0
    for num in lst:
        total += num
    return total
```

```
def find_average(lst):
    return find_sum(lst) / len(lst)

def main():
    while True:
        print("\nList Operations")
        print("1. Perform Operations")
        print("2. Exit")
        choice = input("Choose an option (1-2): ")

        if choice == "2":
            print("Exiting the list operations program. Goodbye!")
            break

        numbers = list(map(float, input("Enter a list of numbers
separated by spaces: ").split()))
        print("Max:", find_max(numbers))
        print("Min:", find_min(numbers))
        print("Sum:", find_sum(numbers))
        print("Average:", find_average(numbers))

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

~/VSCode/Python | main ?4
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-3.py"

List Operations
1. Perform Operations
2. Exit
Choose an option (1-2): 1
Enter a list of numbers separated by spaces: 1 2 3 4 5 6 7 8 9 0
Max: 9.0
Min: 0.0
Sum: 45.0
Average: 4.5

List Operations
1. Perform Operations
2. Exit
Choose an option (1-2): 2
Exiting the list operations program. Goodbye!

~/VSCode/Python | main ?5
```

3.4

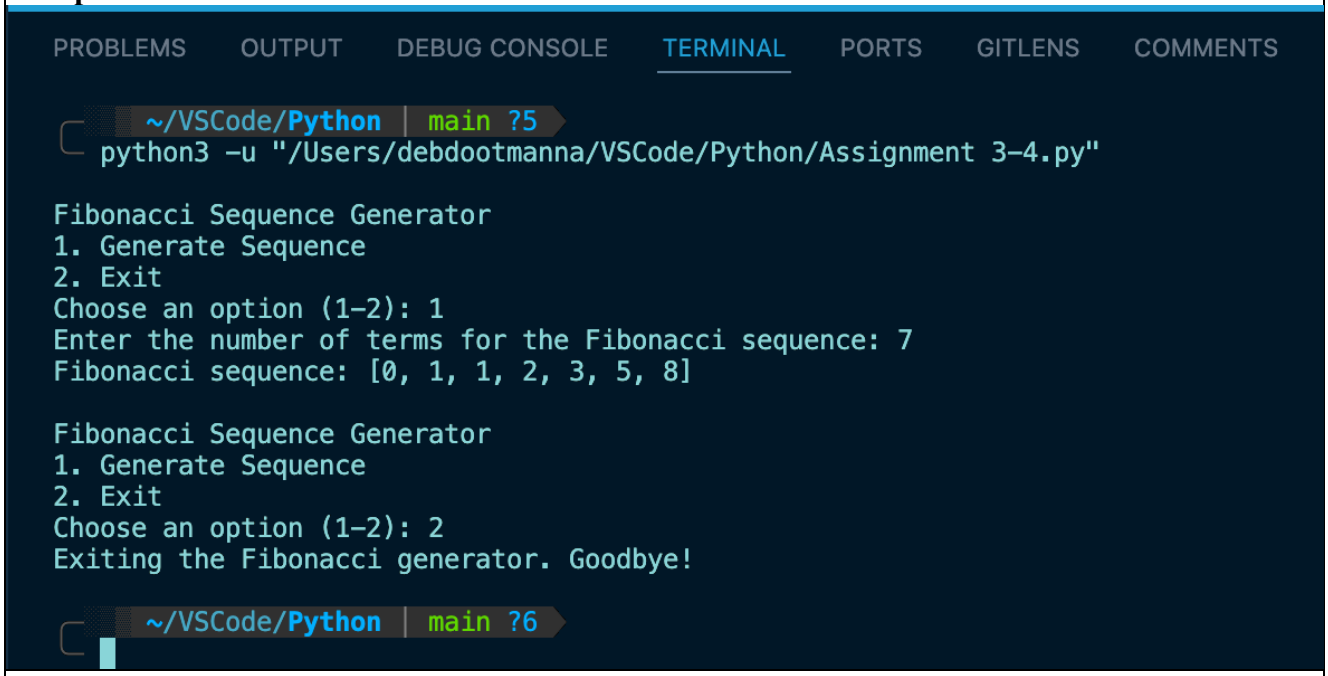
Aim: Write a function that generates the Fibonacci sequence up to a given number of terms using a for loop.

Code:

```python
def fibonacci(n):
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

def main():
    while True:
        print("\nFibonacci Sequence Generator")
        print("1. Generate Sequence")
        print("2. Exit")
        choice = input("Choose an option (1-2): ")

        if choice == "2":
            print("Exiting the Fibonacci generator. Goodbye!")
            break

        terms = int(input("Enter the number of terms for the Fibonacci
```

```
sequence: "))
        print("Fibonacci sequence:", fibonacci(terms))

if __name__ == "__main__":
    main()
```

Output Screenshot:



3.5

Aim: Write functions to add, update, and delete key-value pairs in a dictionary, merge two dictionaries, and display the dictionary contents using loops.

Code:

```
def add_key_value(dictionary, key, value):
    dictionary[key] = value

def update_key_value(dictionary, key, value):
    if key in dictionary:
        dictionary[key] = value
    else:
        print("Key not found")

def delete_key_value(dictionary, key):
    if key in dictionary:
        del dictionary[key]
    else:
        print("Key not found")

def merge_dictionaries(dict1, dict2):
    return {**dict1, **dict2}

def display_dictionary(dictionary):
```

```
    for key, value in dictionary.items():
        print(f"{key}: {value}")

def main():
    my_dict = {}
    while True:
        print("\nDictionary Operations")
        print("1. Add Key-Value")
        print("2. Update Key-Value")
        print("3. Delete Key-Value")
        print("4. Merge Dictionaries")
        print("5. Display Dictionary")
        print("6. Exit")
        choice = input("Choose an option (1-6): ")

        if choice == "6":
            print("Exiting the dictionary operations program.
Goodbye!")
            break

        if choice == "1":
            key = input("Enter key: ")
            value = input("Enter value: ")
            add_key_value(my_dict, key, value)
        elif choice == "2":
            key = input("Enter key: ")
            value = input("Enter new value: ")
            update_key_value(my_dict, key, value)
        elif choice == "3":
            key = input("Enter key to delete: ")
            delete_key_value(my_dict, key)
        elif choice == "4":
            new_dict = {}
            while True:
                key = input("Enter key for new dictionary (or 'done' to
finish): ")
                if key == "done":
                    break
                value = input(f"Enter value for {key}: ")
                new_dict[key] = value
            my_dict = merge_dictionaries(my_dict, new_dict)
        elif choice == "5":
            display_dictionary(my_dict)
        else:
```

```
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

~/VSCode/Python | main ?6
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-5.py"

Dictionary Operations
1. Add Key-Value
2. Update Key-Value
3. Delete Key-Value
4. Merge Dictionaries
5. Display Dictionary
6. Exit
Choose an option (1-6): 1
Enter key: 13
Enter value: 12

Dictionary Operations
1. Add Key-Value
2. Update Key-Value
3. Delete Key-Value
4. Merge Dictionaries
5. Display Dictionary
6. Exit
Choose an option (1-6): 2
Enter key: 13
Enter new value: 15

Dictionary Operations
1. Add Key-Value
2. Update Key-Value
3. Delete Key-Value
4. Merge Dictionaries
5. Display Dictionary
6. Exit
Choose an option (1-6): 5
13: 15

Dictionary Operations
1. Add Key-Value
2. Update Key-Value
3. Delete Key-Value
4. Merge Dictionaries
5. Display Dictionary
6. Exit
Choose an option (1-6): 3
Enter key to delete: 13

Dictionary Operations
1. Add Key-Value
2. Update Key-Value
3. Delete Key-Value
4. Merge Dictionaries
5. Display Dictionary
6. Exit
Choose an option (1-6): 6
Exiting the dictionary operations program. Goodbye!

~/VSCode/Python | main ?7
```

3.6

Aim: Write a program to create a simple to-do list application that allows users to add, remove, and view tasks.

Code:

```
todo_list = []

def add_task(task):
    todo_list.append(task)
    print(f"Task '{task}' added.")

def remove_task(task):
    if task in todo_list:
        todo_list.remove(task)
        print(f"Task '{task}' removed.")
    else:
        print(f"Task '{task}' not found.")

def view_tasks():
    if todo_list:
        print("Your To-Do List:")
        for i, task in enumerate(todo_list, 1):
            print(f"{i}. {task}")
    else:
        print("Your to-do list is empty.")

def main():
    while True:
        print("\nTo-Do List Application")
        print("1. Add Task")
        print("2. Remove Task")
        print("3. View Tasks")
        print("4. Exit")
        choice = input("Choose an option (1-4): ")

        if choice == "4":
            print("Exiting the to-do list application. Goodbye!")
            break

        if choice == "1":
            task = input("Enter the task to add: ")
            add_task(task)
        elif choice == "2":
            task = input("Enter the task to remove: ")
            remove_task(task)
        elif choice == "3":
            view_tasks()
        else:
```

```
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

~/VSCode/Python | main ?7
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-6.py"

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 1
Enter the task to add: Kill Terrorists
Task 'Kill Terrorists' added.

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 1
Enter the task to add: Recreate 911
Task 'Recreate 911' added.

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 3
Your To-Do List:
1. Kill Terrorists
2. Recreate 911

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 2
Enter the task to remove: Recreate 911
Task 'Recreate 911' removed.

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 3
Your To-Do List:
1. Kill Terrorists

To-Do List Application
1. Add Task
2. Remove Task
3. View Tasks
4. Exit
Choose an option (1-4): 4
Exiting the to-do list application. Goodbye!

~/VSCode/Python | main ?8
```

3.7

Aim: Write a program that accepts a list of numbers and returns a new list containing only the even numbers.

Code:

```
def filter_even_numbers(numbers):
    return [num for num in numbers if num % 2 == 0]

def main():
    while True:
        print("\nEven Number Filter")
        print("1. Filter Even Numbers")
        print("2. Exit")
        choice = input("Choose an option (1-2): ")

        if choice == "2":
            print("Exiting the even number filter. Goodbye!")
            break

        numbers = list(map(int, input("Enter a list of numbers
separated by spaces: ").split()))
        print("Even numbers:", filter_even_numbers(numbers))

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

~/VSCode/Python | main ?8
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-7.py"

Even Number Filter
1. Filter Even Numbers
2. Exit
Choose an option (1-2): 1
Enter a list of numbers separated by spaces: 1 2 3 4 5 6 7 8 9 0
Even numbers: [2, 4, 6, 8, 0]

Even Number Filter
1. Filter Even Numbers
2. Exit
Choose an option (1-2): 2
Exiting the even number filter. Goodbye!

~/VSCode/Python | main ?9
```

3.8

Aim: Write a program that finds the largest and smallest numbers in a list without using built-in functions like max() and min()

Code:

```python
def find_largest_smallest(numbers):
    if not numbers:
        return None, None
    largest = smallest = numbers[0]
    for num in numbers:
        if num > largest:
            largest = num
        if num < smallest:
            smallest = num
    return largest, smallest

def main():
    while True:
        print("\nLargest and Smallest Number Finder")
        print("1. Find Largest and Smallest")
        print("2. Exit")
        choice = input("Choose an option (1-2): ")

        if choice == "2":
```

```
            print("Exiting the program. Goodbye!")
            break

        numbers = list(map(float, input("Enter a list of numbers
separated by spaces: ").split()))
        largest, smallest = find_largest_smallest(numbers)
        print("Largest:", largest)
        print("Smallest:", smallest)

if __name__ == "__main__":
    main()
```

Output Screenshot:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

~/VSCode/Python | main ?9
python3 -u "/Users/debdootmanna/VSCode/Python/Assignment 3-8.py"

Largest and Smallest Number Finder
1. Find Largest and Smallest
2. Exit
Choose an option (1-2): 1
Enter a list of numbers separated by spaces: 12 34 96 45 72 38 44
Largest: 96.0
Smallest: 12.0

Largest and Smallest Number Finder
1. Find Largest and Smallest
2. Exit
Choose an option (1-2): 2
Exiting the program. Goodbye!

~/VSCode/Python | main ?10
```

Conclusion/Summary:

This assignment provided a comprehensive exploration of fundamental Python programming concepts, including functions, user input, loops, conditionals, and data structures like lists and dictionaries. By implementing a variety of programs—such as a basic calculator, factorial calculator, list operations, Fibonacci sequence generator, dictionary operations, to-do list application, even number filter, and largest/smallest number finder—we gained hands-on experience in solving real-world problems using Python. Each program was designed to be user-friendly, with clear prompts and an option to exit, ensuring a smooth and interactive experience. This assignment not only reinforced core programming principles but also demonstrated the versatility and power of Python in building practical applications. Moving forward, these foundational skills will serve as a strong base for tackling more complex programming challenges.

Student Signature & Date	**Marks:**	**Evaluator Signature & Date**