***Python Assignment***

***Code Report***

This report provides an explanation of the Python programs I wrote as part of the assignment. Each question covers a specific topic, such as arithmetic operations, comparison, logical operations, string manipulation, and list operations. The programs were written using basic Python syntax and built-in functions.

**Question 1: Arithmetic Operators**

In this exercise, the task was to take two integers from the user and perform various arithmetic operations: addition, subtraction, multiplication, division, modulus, exponentiation, and floor division.

**Code:**

*#Question 1*

x=int(input("Enter a number: ")) *#takes first input from user(integer)*

y=int(input("Enter another number: ")) *#takes another integer as input from user*

*#addition*

print(f"The sum of {x} and {y} =", x+y) *#used + opperator*

*#subtraction*

print(f"The difference of {x} and {y} =", x-y) *#used - opperator*

*#multiplication*

print(f"The product of {x} and {y} =", x\*y) *#used \* operator*

*#divison*

print(f"The quotient of {x} by {y} =", x/y) *#used / operator for float answer*

*#remainder*

print(f"The remainder of {x} by {y} =", x%y) *#used the % operator*

*#exponentiation*

print(f"The answer of {x} to the power of {y} is=", x\*\*y) *#used the \*\* operator*

*#floor division*

print(f"The floor division quotient of {x} by {y} is=", x//y) *#used // operator to get absolute answer*

**Code Explanation:**

The program first takes two integers x and y as input.

The following operators were used:

* + For addition
* - For subtraction
* \* For multiplication
* / For division (returns float)
* % for finding the remainder
* \*\* for exponentiation (raising x to the power of y)
* // for floor division (returns an integer by removing decimal part).

Ex**ample:**

For inputs x=10 and y=3, the results would be:

Sum: 13

Difference: 7

Product: 30

Quotient: 3.33

Remainder: 1

Exponentiation: 1000

Floor Division: 3

**This question reinforced the usage of Python’s arithmetic operators**​

**Question 2: Comparison Operators**

The task here was to take two numbers from the user and compare them to see which one is greater, whether they are equal, or if the first number is less than or equal to the second.

**Code:**

*#Question 2*

x=int(input("Enter a number: "))

y=int(input("Enter another number: "))

*if* x>y: *#conditional statement with condition as x is greater than y*

    print(f"{x} is greater than {y}")

*if* x==y: *#conditional statement with condition as x is equal to y*

    print(f"{x} is equal to {y}")

*if* x<=y: *#conditional statement with condition as x is less than or equal to y*

    print(f"{x} is less than or equal to {y}")

**Code Explanation:**

The program compares the two input numbers using:

* > to check if the first number is greater than the second.
* == to check if the numbers are equal.
* <= to check if the first number is less than or equal to the second.

**Example:**

If x=5 and y=10, the program prints that x is less than or equal to y since 5 is less than 10.

**Question 3: Logical Operators**

This exercise involved taking three Boolean values as input and using logical operators (and, or, and not) to evaluate the conditions.

**Code:**

*#Question 3*

x=bool(input("Enter 1 for True or 0 for False: ")) *#takes boolean values as inputs*

y=bool(input("Enter 1 for True or 0 for False: "))  *#takes boolean values as inputs*

z=bool(input("Enter 1 for True or 0 for False: "))   *#takes boolean values as input*

and\_operator=x and y and z *#the  and operator is used to check if all the conditions are true*

print("The result of the AND operator is: ", and\_operator)

or\_operator=x or y or z  *#the or operator is used to check if any of the conditions are true*

print("The result of the OR operator is: ", or\_operator)

not\_operator\_x= not  x  *#the not operator is used to check if the condition is false*

print("The result of the NOT operator for x is: ", not\_operator\_x)

not\_operator\_y=not y   *#the not operator is used to check if the condition is false*

print("The result of the NOT operator for y is: ", not\_operator\_y)

not\_operator\_z=not z    *#the not operator is used to check if the condition is false*

print("The result of the NOT operator for z is: ", not\_operator\_z)

**Code Explanation:**

* The AND operator checks if all the Boolean values are True.
* The OR operator checks if at least one value is True.
* The NOT-operator negates the Boolean value, flipping True to False and vice versa.

**Example:**

For x=True, y=False, and z=True, the results would be:

AND operator result: False

OR operator result: True

NOT operator result for x: False

**Question 4: String Manipulation**

This exercise required taking a string input and performing operations like finding its length, accessing specific characters, reversing the string, and changing the case.

**Code:**

*#Question 4*

x=input("Enter a string: ")

print(len(x)) *#print the length of the string*

print(x[0])  *# prints the first character of the string*

print(x[::-1]) *#prints the string backwards*

print(x.upper) *#prints the upper case version of the string*

print(x.lower) *#prints the lower case version of the string*

**Code Explanation:**

* len(x) calculates the length of the string.
* x[0] accesses the first character of the string.
* x[::-1] reverses the string.
* x.upper() and x.lower() convert the string to uppercase and lowercase, respectively.

**Example:**

For input x="Hello", the output would be:

Length: 5

First Character: H

Reversed String: olleH

Uppercase: HELLO

Lowercase: hello

**This exercise demonstrated string slicing and the use of string methods to manipulate text​.**

**Question 5: String Formatting**

The goal here was to ask for the user's name and age and display a formatted message.

**Code:**

*#Question 5*

x=int(input("Enter your age: ")) *#takes integer input from user*

y=input("Enter your Name")  *#takes string input from user*

print(f"Hello {y}, you are {x} years old")  *#prints the output in the specified format*

**Code Explanation:**

* The input() function collects the name and age.
* Using an f-string, the output is formatted like: f"Hello {name}, you are {age} years old".

**Example:**

For name="Alice" and age=25, the output would be:

Hello Alice, you are 25 years old.

**This exercise focused on creating user-friendly outputs using string formatting.**

**Question 6: Substring Search**

In this exercise, the program takes a sentence and a word as input and checks if the word exists in the sentence, then outputs the index of the word if found.

**Code:**

*#Question 6*

x=input("Enter a sentence: ")

y=input("Enter a word to search for in the sentence: ")

*if* y in x:

    z=x.index(y)  *#gives  the index of the first occurrence of the word*

    print(f"The word '{y}' is in the sentence and is positioned at index {z}")

*else*:

    print(f"The word '{y}' is not in the sentence")

**Code Explanation:**

* The in keyword checks if the word exists in the sentence.
* If found, x.index(y) returns the index of the word’s first occurrence.

**Example:**

For sentence="Python is great" and word="great", the output would be:

The word 'great' is in the sentence and is positioned at index 10

**This highlights the use of substring search and basic string operations​.**

**Question 7: List Operations**

The task was to create a list of five numbers, then calculate the sum, largest number, and smallest number.

**Code:**

*#Question 7*

a=[ ] *#creates an empty list*

*for* i *in* range(5):

    x=int(input(f"Enter the {i+1} number: ")) *#asks for 5 numbers*

    a.append(x) *#the append function adds an element to the end of the list*

    print("The list is: ", a)  *#prints the list after each number is added*

print("The sum of all the terms is", sum(a)) *#prints the sum oif all numbers in the list*

print("The largest number in the list is", max(a)) *#prints the largest number in the list*

print("The smallest number in the list is", min(a)) *#prints the smallest number in the list*

**Code Explanation:**

* A loop is used to take 5 inputs, which are added to the list using .append().
* sum(), max(), and min() are used to find the sum, maximum, and minimum of the list.

**Example:**

For the list [3, 7, 1, 8, 5], the output would be:

Sum: 24

Largest number: 8

Smallest number: 1

**This demonstrated the power of list functions in Python.**

**Question 8: List Manipulation**

This exercise asked for a list of fruits and allowed for adding and removing elements from the list.

**Code:**

*#Question 8*

a=[ ] *#creates an empty list*

*for* i *in* range(5):

    x=input(f"Enter a list of fruits you want {i}: ") *#asks for user input*

    a.append(x) *#the append function adds an element to the end of the list*

    print("The list is: ", a)  *#prints the list after each number is added*

b=input("Enter a fruit of choice: ") *#asks for 1 fruit of choice from the user*

a.append(b) *#the list is appended with the fruit of choice*

print(a.pop(1)) *# .pop(1) removes the 2nd item from the list*

print("The final list is: ",a)

**Code Explanation:**

* The .append() method adds a fruit to the list.
* The .pop(1) method removes the second item from the list (index starts at 0).

**Example:**

For an initial list ["Apple", "Banana", "Cherry"] and adding Mango:

Final List: ["Apple", "Cherry", "Mango"]

**The program manipulates the list as per user input**

**Question 9: Sorting a List**

The task here was to sort a list of five numbers in both ascending and descending order.

**Code:**

*#Question 9*

x = [ ]

*for* i *in* range(5):

    a = int(input(f"Enter number {i+1}: ")) *#Ask the user to input a list of 5 numbers*

    x.append(a)

*# Sort the list in ascending order*

b = sorted(x)

print("List in ascending order:", b)

*# Sort the list in descending order*

c = sorted(x, *reverse*=True)

print("List in descending order:", c)

**Code Explanation:**

* sorted(x) sorts the list in ascending order.
* sorted(x, reverse=True) sorts the list in descending order.

**Example:**

For the list [8, 3, 5, 1, 9], the output would be:

Ascending Order: [1, 3, 5, 8, 9]

Descending Order: [9, 8, 5, 3, 1]

**This reinforced the concept of sorting lists​.**

**Question 10: List Slicing**

Given a predefined list of 10 numbers, this task required slicing the list to display different parts of it.

**Code:**

*#Question 10*

x=[1,2,3,4,5,6,7,8,9,10] *#predefined list*

print("The first 5 elements of the list are:", x[:5])

print("The last 5 elements of the list are:", x[-5:])

print("The numbers from index 2 to index 7 are:", x[2:8])

**Code Explanation:**

* x[:5] retrieves the first 5 elements.
* x[-5:] retrieves the last 5 elements.
* x[2:8] retrieves elements between index 2 and 7.

**Example:**

For the list [1,2,3,4,5,6,7,8,9,10], the output would be:

First 5 elements: [1, 2, 3, 4, 5]

Last 5 elements: [6, 7, 8, 9, 10]

Index 2 to 7: [3, 4, 5, 6, 7, 8]

**This exercise helped understand how to extract parts of a list.**

**Bonus Question: Nested Lists**

In this bonus question, we gathered names and scores of three students, then calculated their average scores.

**Code:**

*#Bonus Question: Question 11*

x = [ ] *#Empty list to store students' data*

*# Taking input for 3 students*

*for* i *in* range(3):

    y = input(f"Enter the name of student {i + 1}: ")  *#Taking input for name*

    scores = [ ]   *#Empty list to store scores of each student*

*for* j *in* range(3):  *# 3 subjects*

        score = float(input(f"Enter the score for subject {j + 1}: "))

        scores.append(score)

    x.append([y, scores])

*# Calculating and displaying the average score for each student*

print("\nStudent's Name and Average Score:")

*for* student *in* x:  *# Iterate over x instead of y*

    name = student[0]

    scores = student[1]

    average = sum(scores) / len(scores)  *# Calculate average score*

    print(f"{name}'s average score: {average:.2f}")

**Code Explanation:**

* A nested list was used to store each student's name and their scores in three subjects.
* The average score was calculated by summing the scores and dividing by the number of subjects.

**Example:**

For student="Alice" with scores [85, 90, 88], the average would be:

Alice's average score: 87.67

**This exercise demonstrated the use of nested lists for handling structured data.**

**Conclusion:**

This assignment reinforced several key Python concepts, including arithmetic and comparison operators, string manipulation, and list operations. The exercises allowed me to practice problem-solving and understand how to use Python’s built-in functions efficiently.