Set 1

1. Data Types

Problem 1: Identify the data types of the following variables:

x = 10
y = 3.14
z = "Hello, Python!"
a = True
b = [1, 2, 3]
c = (4, 5, 6)
d = {7, 8, 9}
e = {"name": "Alice", "age": 25}

Write a Python script to print the data types of each variable using the `type()` function.

Problem 2: Perform operations with mixed data types and explain the result:

a = 5
b = "10"
c = 2.5

- Add `a` and `c`.
- Concatenate `b` with `" is a number"`.
- Try adding `a` and `b` (What happens? Why?).

---

2. Type Conversion

Problem 3: Convert the following data types:

- Convert an integer `x = 15` to a string, float, and boolean.
- Convert a float `y = 3.99` to an integer and a string.
- Convert a string `z = "123"` into an integer and a float.

Problem 4: Write a program that accepts a user's input as a string and converts it to:

- Integer
- Float
- Boolean
  Print the converted values.

---

3. Loops

Problem 5: Use a `for` loop to print all the even numbers between 1 and 50.

Problem 6: Write a program to calculate the factorial of a number using a `while` loop.
Example: Input = `5`, Output = `120` (because 5!=5×4×3×2×1=1205! = 5 \times 4 \times 3 \times 2 \times 1 = 1205!=5×4×3×2×1=120)

Problem 7: Write a Python program to create the following pattern using nested loops:

```
*
**
***
****
*****
```

Problem 8: Iterate through the following dictionary using a loop and print its keys and values:

student = {"name": "John", "age": 20, "grade": "A"}

Problem 9: Create a list of numbers from 1 to 10. Use a loop to create a new list containing the squares of these numbers.
Example: Input = `[1, 2, 3]`, Output = `[1, 4, 9]`

Problem 10: Write a program that uses a loop to reverse a string.
Example: Input = `"Python"`, Output = `"nohtyP"`

Bonus Challenges

Challenge 1: Write a program that counts the number of vowels in a given string.
Example: Input = `"Hello, Python!"`, Output = `4`

Challenge 2: Use a loop to check if a given number is a prime number.
Example: Input = `7`, Output = `"Prime"`

Set 2

1. String Methods

Problem 1: Write a program that takes a string input and performs the following:

- Convert the string to uppercase.
- Replace all spaces with underscores (_).
- Check if the string ends with a specific substring (e.g., `"ing"`).

Problem 2: Given the string:

text = " Python is Amazing! "

- Remove the leading and trailing whitespaces.
- Count the occurrences of the letter `"i"`.
- Split the string into words.

Problem 3: Write a program that reverses a string using slicing and the `join()` method.
Example: Input = `"Hello"`, Output = `"olleH"`

---

2. Number Methods

Problem 4: Use the following number:

num = -25.678

- Find its absolute value.
- Round it to 1 decimal place.
- Convert it into an integer.

Problem 5: Write a program to check if a given number is even or odd using the modulo operator and print the result.

---

3. Boolean Methods

Problem 6: Write a program that uses the `any()` and `all()` functions on the following list:

bool_list = [True, False, True, False]

- Use `any()` to check if at least one element is `True`.
- Use `all()` to check if all elements are `True`.

---

4. List Methods

Problem 7: Given the list:

numbers = [5, 3, 8, 6, 2]

- Append the number `10`.
- Sort the list in ascending order.
- Remove the smallest number.

Problem 8: Write a program to:

- Create a list of 5 names.
- Use `index()` to find the position of a specific name.
- Use slicing to print the first three names.

---

5. Tuple Methods

Problem 9: Given the tuple:

items = (10, 20, 30, 40, 50)

- Find the index of `30`.
- Count how many times `20` appears in the tuple.
- Convert the tuple into a list and add a new item.

---

6. Set Methods

Problem 10: Write a program to:

- Create two sets: `A = {1, 2, 3}` and `B = {3, 4, 5}`.
- Find the union of the sets.
- Find the intersection of the sets.
- Find the difference (`A - B`).

Problem 11: Remove all duplicates from the following list using a set:

nums = [1, 2, 2, 3, 4, 4, 5]

---

7. Dictionary Methods

Problem 12: Given the dictionary:

student = {"name": "Alice", "age": 20, "grade": "A"}

- Add a new key-value pair for `"subject": "Math"`.
- Update the `"grade"` to `"A+"`.
- Use a loop to print all keys and values.

Problem 13: Write a program that:

- Creates a dictionary with 5 items.
- Deletes an item using the `pop()` method.
- Clears all items using the `clear()` method.

8. Conditional Statements (if-elif-else)

Problem 14: Write a program to check if a number is:

- Positive
- Negative
- Zero

Problem 15: Write a program to calculate the grade of a student based on marks:

- Marks >= 90: Grade A
- Marks >= 75: Grade B
- Marks >= 50: Grade C
- Else: Grade F

---

9. While Loop

Problem 16: Write a program to calculate the sum of numbers from 1 to 100 using a `while` loop.

Problem 17: Write a program that takes user input repeatedly until the user enters `"quit"`. Print all the inputs received.

Problem 18: Use a `while` loop to generate the following pattern:

1
12
123
1234
12345

Bonus Challenges

Challenge 1: Write a program to check if a string is a palindrome (reads the same forwards and backwards).
Example: Input = `"radar"`, Output = `"Palindrome"`

Challenge 2: Write a program to count the frequency of each character in a string using a dictionary.
Example: Input = `"hello"`, Output = `{'h': 1, 'e': 1, 'l': 2, 'o': 1}`

Set 3

1. Print Numbers

Write a program to print all numbers from 1 to 10 using a `for` loop and the `range()` function.

---

2. Print Even Numbers

Write a program to print all even numbers between 1 and 20.

---

3. Reverse Range

Use a `for` loop to print numbers from 10 down to 1 in reverse order.

---

4. Sum of Numbers

Write a program to calculate the sum of all numbers from 1 to 50 using a `for` loop.

---

5. Multiplication Table

Write a program that generates the multiplication table of a given number (e.g., 5) using a `for` loop and `range()`.

---

6. Skip Counting

Use a `for` loop to print numbers from 1 to 50, skipping every third number (e.g., 1, 4, 7, ...).

---

7. Square of Numbers

Write a program that prints the squares of all numbers from 1 to 10.

8. Sum of Odd Numbers

Write a program to calculate the sum of all odd numbers between 1 and 50 using a `for` loop and `range()`.

---

9. Nested Range

Use nested `for` loops to generate the following pattern:

```
1
12
123
1234
12345
```

---

10. Prime Numbers

Write a program to print all prime numbers between 1 and 50 using a `for` loop and `range()`. (Hint: Use an inner `for` loop to check divisibility of each number.)