Q1. Create one variable containing following type of data:

(i) string

(ii) list

(iii) float

(iv) tuple

Ans- (i) string

my\_string = "Hello, World!"

(ii) list

my\_list = [1, 2, 3, 4, 5]

(iii) float

my\_float = 3.14

(iv) tuple

my\_tuple = (1, "apple", 2.5, True)

Q2. Given are some following variables containing data:

(i) var1 = ‘ ‘

(ii) var2 = ‘[ DS , ML , Python]’

(iii) var3 = [ ‘DS’ , ’ML’ , ‘Python’ ]

(iv) var4 = 1.

What will be the data type of the above given variable.

Ans- (i) Its data type is str.

(ii) Its data type is str.

(iii) Its data type is list.

(iv) Its data type is float.

Q3. Explain the use of the following operators using an example:

(i) /

(ii) %

(iii) //

(iv) \*\*

Ans-

(i) / (Division Operator): The division operator is used to perform division between two numbers. It returns the quotient of the division as a floating-point number.

Example:

a = 10

b = 3

result = a / b

print(result) # Output: 3.3333333333333335

In the above example, a is divided by b using the division operator /. The result is a floating-point number, 3.3333333333333335.

(ii) % (Modulo Operator): The modulo operator is used to calculate the remainder of a division operation. It returns the remainder as the result.

Example:

a = 10

b = 3

result = a % b

print(result) # Output: 1

In the above example, a is divided by b using the modulo operator %. The remainder of the division is 1.

(iii) // (Floor Division Operator): The floor division operator is used to perform division between two numbers and return the quotient as an integer, rounded down to the nearest whole number (towards negative infinity).

Example:

a = 10

b = 3

result = a // b

print(result) # Output: 3

In the above example, a is divided by b using the floor division operator //. The quotient of the division is 3, without any decimal places.

(iv) \*\* (Exponentiation Operator): The exponentiation operator is used to raise a number to the power of another number.

Example:

a = 2

b = 3

result = a \*\* b

print(result) # Output: 8

In the above example, a is raised to the power of b using the exponentiation operator \*\*. The result is 8 because 2 raised to the power of 3 is equal to 8.

Q4. Create a list of length 10 of your choice containing multiple types of data. Using for loop print the element and its data type.

Ans- my\_list = [1, 'apple', 3.14, True, [1, 2, 3], 'banana', (4, 5, 6), {'name': 'John', 'age': 25}, None, 5.6]

for element in my\_list= print(element, type(element))

1 <'int'>

apple <'str'>

3.14 <'float'>

True <’bool'>

[1, 2, 3] <'list'>

banana <'str'>

(4, 5, 6) <'tuple'>

{'name': 'John', 'age': 25} <'dict'>

None <'NoneType'>

5.6 <'float'>

Q5. Using a while loop, verify if the number A is purely divisible by number B and if so then how many times it can be divisible.

Ans- A = 100

B = 5

count = 0

while A % B == 0:

A = A / B

count += 1

print("Number of times A is divisible by B:", count)

Number of times A is divisible by B: 2

Q6. Create a list containing 25 int type data. Using for loop and if-else condition print if the element is divisible by 3 or not.

Ans- my\_list = [8, 15, 21, 9, 6, 12, 17, 30, 4, 27, 5, 18, 11, 20, 13, 24, 7, 3, 14, 19, 22, 10, 16, 2, 25]

for element in my\_list:

if element % 3 == 0:

print(f"{element} is divisible by 3")

else:

print(f"{element} is not divisible by 3")

8 is not divisible by 3

15 is divisible by 3

21 is divisible by 3

9 is divisible by 3

6 is divisible by 3

12 is divisible by 3

17 is not divisible by 3

30 is divisible by 3

4 is not divisible by 3

27 is divisible by 3

5 is not divisible by 3

18 is divisible by 3

11 is not divisible by 3

20 is not divisible by 3

13 is not divisible by 3

24 is divisible by 3

7 is not divisible by 3

3 is divisible by 3

14 is not divisible by 3

19 is not divisible by 3

22 is not divisible by 3

10 is not divisible by 3

16 is not divisible by 3

2 is not divisible by 3

25 is not divisible by 3

Q7. What do you understand about mutable and immutable data types? Give examples for both showing this property.

Ans- Examples of immutable data types in Python include:

Numbers (integers, floats)

Strings

Tuples

Example of an immutable data type (string)-

my\_string = "Hello"

my\_string += " World"

print(my\_string) # Output: Hello World

Examples of mutable data types in Python include:

Lists

Dictionaries

Sets

Example of a mutable data type (list)-

my\_list = [1, 2, 3]

my\_list.append(4)

print(my\_list) # Output: [1, 2, 3, 4]