**Python Advanced Assignment 13**

Q1. Can you create a programme or function that employs both positive and negative indexing? Is

there any repercussion if you do so?

Ans-) Yes, you can create a program or function that employs both positive and negative indexing in Python. There is no repercussion in doing so. Positive indexing starts from 0, while negative indexing starts from -1.

Q2. What is the most effective way of starting with 1,000 elements in a Python list? Assume that all

elements should be set to the same value.

Ans-) The most effective way of starting with 1,000 elements in a Python list, all set to the same value, is to use a list comprehension:

my\_list = [initial\_value] \* 1000

Q3. How do you slice a list to get any other part while missing the rest? (For example, suppose you

want to make a new list with the elements first, third, fifth, seventh, and so on.)

Ans-) To slice a list and get only the elements with odd indices, you can use the following syntax:

my\_list[::2]

Q4. Explain the distinctions between indexing and slicing.

Ans-) Indexing is used to retrieve a specific element from a list, while slicing is used to retrieve a specific subset of elements from a list.

Q5. What happens if one of the slicing expression’s indexes is out of range?

Ans-) If one of the slicing expression's indexes is out of range, Python returns an empty list.

Q6. If you pass a list to a function, and if you want the function to be able to change the values of the

list—so that the list is different after the function returns—what action should you avoid?

Ans-) If you want a function to be able to change the values of a list, you should avoid passing the list as an argument to a function and then reassigning the variable inside the function. This is because the reassignment creates a new reference to a new object, and the original list is left unchanged. Instead, you should modify the original list in-place using its methods or by directly accessing its elements.

Q7. What is the concept of an unbalanced matrix?

Ans-) An unbalanced matrix is a matrix where the number of elements in each row is not equal. It is also known as a jagged array.

Q8. Why is it necessary to use either list comprehension or a loop to create arbitrarily large matrices?

Ans-) It is necessary to use list comprehension or a loop to create arbitrarily large matrices because Python does not provide built-in support for creating matrices of arbitrary size. Additionally, matrices are usually implemented as lists of lists, which requires iteration to generate.