**Q1. Can you create a programme or function that employs both positive and negative indexing? Is there any repercussion if you do so?**

Yes, it is possible to use both positive and negative indexing in a Python program. Positive indexing is used to access elements in a list or array starting from the beginning of the sequence, whereas negative indexing is used to access elements starting from the end of the sequence.

For example, in a list my\_list = [10, 20, 30, 40, 50], the elements can be accessed using positive and negative indexing as follows:

# Positive indexing

my\_list[0] # returns 10

my\_list[1] # returns 20

my\_list[2] # returns 30

my\_list[3] # returns 40

my\_list[4] # returns 50

# Negative indexing

my\_list[-1] # returns 50

my\_list[-2] # returns 40

my\_list[-3] # returns 30

my\_list[-4] # returns 20

my\_list[-5] # returns 10

**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.**

One effective way to start with 1,000 elements in a Python list and set all the elements to the same value is to use the **list()** function and the **\*** operator as follows:

*# Start with an empty list and set all elements to 0*

***my\_list = [0] \* 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.)**

To slice a list in Python and create a new list with certain elements while excluding the others, you can use the **list[start:end:step]** syntax. This allows you to specify a starting index, an ending index, and a step size to determine which elements to include in the new list.

**Q4. Explain the distinctions between indexing and slicing.**

In Python, indexing and slicing are ways of accessing parts of sequences (strings, lists, etc.). Indexing is used to access individual elements of a sequence, while slicing allows you to access a sub-sequence or a range of elements within a sequence.

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

If one of the indexes in a slicing expression is out of range, Python will simply return a sub-sequence that ends at the end of the original sequence, rather than raising an error.

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?

**Q7. What is the concept of an unbalanced matrix?**

In Python, a matrix is said to be unbalanced if the number of rows and columns are not equal.

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

In Python, you can create a matrix by using a list of lists, where each inner list represents a row of the matrix. However, if you want to create a matrix with a large number of rows and columns, it can be impractical to do so by manually specifying each element of the matrix. In this case, it is more efficient to use either a loop or list comprehension to generate the matrix automatically.