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

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.

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.)

Q4. Explain the distinctions between indexing and slicing.

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

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?

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

Answer:

Q1. Yes, you can use both positive and negative indexing in a program or function that operates on a Python list. There is no repercussion in doing so, as negative indexing is simply a way to index elements from the end of the list.

Q2. One effective way to start with 1,000 elements in a Python list is to use a list comprehension:

my\_list = [0] \* 1000

This creates a list with 1,000 elements, all set to the value 0.

Q3. To slice a list to get every other part, you can use the slice notation with a step of 2:

my\_list[::2]

This will return a new list with every other element from the original list.

Q4. Indexing is the process of accessing a single element of a list by its position in the list. Slicing is the process of accessing a subset of a list by specifying a range of positions in the list.

Q5. If one of the slicing expression's indexes is out of range, a "IndexError: list index out of range" error will be raised.

Q6. If you want the function to be able to change the values of the list, you should avoid reassigning the list parameter to a new list inside the function. Instead, you should modify the list in place using its methods or by assigning new values to specific indices.

Q7. An unbalanced matrix is a matrix in which the number of elements in each row is different.

Q8. It is necessary to use either list comprehension or a loop to create arbitrarily large matrices because they are not fixed-size data structures like arrays in some other programming languages. Python lists are dynamic and can grow or shrink as needed, so you need to use a loop or list comprehension to create the appropriate number of rows and columns in the matrix.