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

Ans: Yes. there are no repercussions

2. 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: start_list = [1 for x in range(1001)]

3. 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.) In Python, how many different ways are there to index a character?

Ans: sliced_list = my_list[::x] where x is the part you want

4. Explain the distinctions between indexing and slicing.

Ans: Indexing is used to obtain individual elements while slicing for sequence of elements.

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

Ans: If index is out of range then it will return an empty list.

6. 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: Always use a return statement, if we want to see the changes in the input list.

7. What is the concept of an unbalanced matrix?

Ans: In an unbalanced matrix, the number of rows is not the same as the number of columns.

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

Ans: List comprehension or a loop helps creation of large matrices easy. It also helps to implement and avoid manual errors and reduces a lot of time as manual feeding avoided.