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

***Ans***:

def substr\_between(string, index1, index2):

if index1 < 0:

index1 = len(string) + index1

if index2 < 0:

index2 = len(string) + index2 + 1

return string[min(index1, index2):max(index1, index2)]

there is no significant repercussion to using both positive and negative indexing in a program or function, if you handle the indices correctly and consistently.

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 to start with 1,000 elements in a Python list, all set to the same value, is to use the multiplication operator \* to create a list with the desired number of elements and then use the list() function to convert the resulting object into a list.

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***:

original\_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

new\_list = original\_list[::2]

The resulting new list will contain the elements 1, 3, 5, 7, 9.

Q4. Explain the distinctions between indexing and slicing.

***Ans***:

Indexing and slicing are both ways to access and retrieve elements from a sequence (such as a list, tuple, or string) in Python, but they have different meanings and behaviours.

Indexing refers to the process of selecting a single element from a sequence using its position (or index) in the sequence.

Slicing, on the other hand, refers to the process of selecting a subset of elements from a sequence using a range of indices.

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

***Ans***:

if one of the slicing expressions' indexes is out of range, Index Error will be raised.

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 that is passed as an argument, you should avoid reassigning the variable that holds the list inside the function.

However, if you reassign the variable that holds the list inside the function, you will create a new local variable that refers to a different object, and any changes made to that object will not affect the original list outside the function:

Q7. What is the concept of an unbalanced matrix?

***Ans***:

An unbalanced matrix is a matrix where the number of rows and columns is not equal. This means that the matrix cannot be square, and there will be some elements in the matrix that have no corresponding element in the other dimension.

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

***Ans***:

To create an arbitrarily large matrix, you need to create a list of the desired number of rows, and then for each row, create a list of the desired number of columns.