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

In Python, you can start indexing from the end of an iterable. This is known as negative indexing. This means you can use both positive and negative indexes to access iterables.

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.

Simple loop with append : my\_list = [] for i in range(50): my\_list.append(0)

Simple loop with += : my\_list = [] for i in range(50): my\_list += [0]

List comprehension: my\_list = [0 for i in range(50)]

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.)

List slicing returns a new list from the existing list. If Lst is a list, then the above expression returns the portion of the list from index ...

Q4. Explain the distinctions between indexing and slicing.

: Indexing is used to obtain individual elements. Slicing: Slicing is used to obtain a sequence of elements. Indexing and Slicing can be be done in Python Sequences types like list, string, tuple, range objects.

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

Ans = The slicing operation doesn't raise an error if both your start and stop indices are larger than the sequence length. This is in contrast to simple indexing—if you index an element that is out of bounds, Python will throw an index out of bounds error. However, with slicing it simply returns an empty sequence.

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 = So passing lists to a function where they're used with different variable names causes them to be altered? Never expected that, but it ...

Q7. What is the concept of an unbalanced matrix?

Ans = Whenever the cost matrix of an assignment problem is not a square matrix, that is, whenever the number of sources is not equal to the number of destinations, the assignment problem is called an unbalanced assignment problem.

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

Ans = List comprehensions apply an arbitrary expression to items in an iterable rather than applying function