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

**Ans : list = [1,2,3,4,5,6,7]**

**Positive indexing : print(list[2:6])**

**Negative Indexing : print(list[-5:-3])**

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 List comprehension is effective way.**

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: We can use third parameter in list which is jump. List[start:end:jump] it will slice the list according to this scenario**

Q4. Explain the distinctions between indexing and slicing.

**Ans: Indexing means referring to an element of an iterable by its position within the iterable. Slicing means getting a subset of elements from an iterable based on their indices.**

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

**Ans: it will throw an error that list index 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?

**Ans: if we pass the list to function it will not affect the global list. It will just print the values in local list. If we want to update the data in global list means we can call with help of global keyword**

Q7. What is the concept of an unbalanced matrix?

**Ans: the number of sources is not equal to the number of destinations, the assignment problem is called an unbalanced assignment**

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

**Ans: because the iteration or list comprehension will iterate the data one by one so that it will easily handle the large matrices**