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 indexing():

string="I live in India"

print(string[2:])

print(string[-2:])

No, there is no repercussion.

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:**

If all values in list will remain same then there is no need to keep 1000 elements in list, just keep 1 element and multiply it with 1000.

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:**

l[0::2]

Q4. Explain the distinctions between indexing and slicing.

**Ans:**

Indexing is basically to get l particular item of particular index, while slicing is to get a particular portion.

l[2] is an example of indexing, while l[2:6] is an example of slicing.

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

**Ans:**

It will not throw any error however it will return items till max/min length.

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:**

We should avoid copy function.

Q7. What is the concept of an unbalanced matrix?

**Ans:**

Whenever the matrix is not a square matrix, is called an unbalanced matrix.

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

**Ans:**

Matrices are arrays in programming language, but in python there is no concept of arrays so it becomes necessary to use either list comprehension or a loop to create arbitrarily matrices.