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

Yes, there are no repercussions as long as it is in sequence

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.

We can use an array or use a list slice command.

a = [1]\*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.)

by using l[::n]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

L[1:::2]

Starting from one it will slice every 2nd odd number.

Q4. Explain the distinctions between indexing and slicing.

The indexing is used as a resource to obtain individual elements and slicing is used to obtain a sequence of those elements.

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

If one of the slicing expression's indexes is out of range then python will generate an error and simply return 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?

Create a mutable list type with indexing that can be later reassigned new value. List indexing: We use the index number of a list item to modify the value associated with that item. If you want to replace one value in a list, the indexing syntax is most appropriate. To replace multiple items in a list that meet a criterion, using list comprehension is a good solution.

Q7. What is the concept of an unbalanced matrix?

A matrix is unbalanced if all cells in the matrix are unbalanced, and a cell of the matrix is unbalanced if the number of cells in that matrix that are adjacent to that cell is strictly lesser than the value written in this cell.

Adjacent cell means cells in the top, down, left, and right cell of each cell if it exists.

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

It helps to build the code more readable and help understand how it works.

List comprehension is fast effective and requires much less effort.