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

Yes we can create a program to check if string is palindrom or not and use positive indexing and negative indexing as follows

def palindrom(s):

i=0

while i <= len(s)/2:

if s[i] != s[-i]:

return False

return True

repercussion of this could be it can make program harder to understand and list can go out of bounds.

**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 [default\_value]\*1000 to create 1,000 elements in a Python list in which all values are same

**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.)**

I can use list comprehension in this case 🡪 [ x for x in list if x%2 != 0]

**Q4. Explain the distinctions between indexing and slicing.**

indexing allows you to access a single element at a specific position within a sequence using its index. On the other hand, slicing enables you to extract a subsequence or a range of elements from a sequence based on a specified range of indices.

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

Python automatically adjusts the out-of-range indices to fit within the valid range, ensuring that the slicing operation does not raise an error and produces the expected results.

**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?**

* Giving index greater than length of list
* Reassigning the elements of list
* Changing the list elements before operating on

Are the action that should be avoided while writing program.

**Q7. What is the concept of an unbalanced matrix?**

The concept of an unbalanced matrix refers to a matrix where the number of elements in each row or column is not equal. In other words, the rows or columns of the matrix have varying lengths.

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

By using list comprehension or loop we can create matrices dynamically of varying sizes. We can perform different operations on elements while creating matrices.