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

Ans=>No we cant.

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=> size =1000

pre\_allocated\_list = [None] \* size

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 step here like print(l[1:10:3])

Q4. Explain the distinctions between indexing and slicing.

Ans=> Indexing is used to obtain individual elements

Slicing is used to obtain a sequence of elements.

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

Ans=>it will print till last element

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 will avoid different indexing of list

Q7. What is the concept of an unbalanced matrix?

Ans=> A matrix is balanced if all cells in the matrix are balanced and a cell of the matrix is balanced if the number of cells in that matrix that are adjacent to that cell is strictly greater than the value written in this cell

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

Ans=>because in list we can use nested list.