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

**Ans:**

**my\_list = [1,2,3,4,5,6,6,7,8,9,10]**def **bi\_index(in\_list,position):** return **in\_list[position]  
print('Positive Indexing ->',bi\_index(my\_list,5))  
print('Negative Indexing ->',bi\_index(my\_list,-1))**

**Positive Indexing -> 6  
Negative Indexing -> 10**

#### 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:**

**start\_list = [1** for **x** in **range(1001)] # Quick Way to Create a List Using List Comprehension  
print(start\_list)**

#### 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:**

**my\_list = [x** for **x** in **range(1,15)]  
print(f'my\_list -> {my\_list}')  
sliced\_list = my\_list[::2]  
print(f'sliced\_list -> {sliced\_list}')**

**my\_list -> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]  
sliced\_list -> [1, 3, 5, 7, 9, 11, 13]**

#### Q4. Explain the distinctions between indexing and slicing ?

**Ans:**

**Indexing is used when we have to work on index level. While slicing are used over a range of items.**

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

**Ans:**

**If start index is out of range then it will return empty entity.**

#### 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 do?

**Ans:**

**Always use return statement, if we want the see the changes in the input list.**

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

**Ans:**

**In Unbalanced Matrix number of rows is not same as number of columns.**

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

**Ans:**

**List comprehension or a Loop helps creation of large matrices easy. it also helps to implemeent and avoid manual errors. it also makes reading code easy.**