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

Ans : In Python, list slicing is a common practice and it is the most used technique for programmers to solve efficient problems. Consider a python list, In-order to access a range of elements in a list, you need to slice a list. One way to do this is to use the simple slicing operator i.e. colon(:)

1. With this operator, one can specify where to start the slicing, where to end, and specify the step. List slicing returns a new list from the existing list.

# Initialize list

Lst = [50, 70, 30, 20, 90, 10, 50]

# Display list

print(Lst[::])

1. Negative Indexes

Index -1 represents the last element and -n represents the first element of the list(considering n as the length of the list). Lists can also be manipulated using negative indexes also.

# Initialize list

Lst = [50, 70, 30, 20, 90, 10, 50]

# Display list

print(Lst[-7::1])

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 : lst = [0] \* 1000

print(listOfStrings1)

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 : newLst = lst[start:stop:step]

Q4. Explain the distinctions between indexing and slicing.

Ans : “Indexing” means referring to an element of an iterable by its position within the iterable. “Slicing” means getting a subset of elements from an iterable based on their indices.

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

Ans : This is in contrast to simple indexing—if you index an element that is out of bounds, Python will throw an index out of bounds error. However, with slicing it simply returns 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?

Ans :

Reassigning a variable inside a function won't change the variable you passed as an argument outside the function. Performing mutable operations on the variable, however, will change it.

Q7. What is the concept of an unbalanced matrix?

Ans : Whenever the cost matrix of an assignment problem is not a square matrix, that is, whenever the number of sources is not equal to the number of destinations, the assignment problem is called an unbalanced assignment problem.

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