**ASSIGNMENT 13**

Q1. Can you create a program or function that employs both positive and negative indexing? Is

there any repercussion if you do so?

Ans: Yes, we can create a program or function that employs both positive and negative indexing and there are no repercussions at all.

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: The most effective way of starting with 1000 elements in python list is a=[1]\*1000, where a is the list containing 1 element thousand times. The second most effective way is list comprehension a=[1 for I in range(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.)

Ans: Suppose we have a list b=[1,2,3,4,5,6,7,8,9,10] then if want to make a new list of [1,3,5,7,9] from this then we can slice the b like this:

b=[1,2,3,4,5,6,7,8,9,10]

c=[]

for i in range(0,len(b),2):

c.append(b[i])

print(c)

Q4. Explain the distinctions between indexing and slicing.

Ans: Indexing is accessing the elements of an iterable, whereas, slicing is creating a new iterable from the existing iterable using indexes.

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

Ans: Slicing expression have three parts a,b,c where a is the starting index of an iterable, b is the ending index of iterable and c is the number of indexes it will skip. Therefore, if we make the a part out of range then the output will be empty string, if we make the ending index out of range then the program will throw an IndexError, and if we make c out of range then all the elements coming in the range jumping c indexes will be in sublist or subiterable.

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 should avoid changing the original list because we may need to use the original list elsewhere, therefore, by creating the copy of the list in the function can help tackle this problem.

Q7. What is the concept of an unbalanced matrix?

Ans: Unbalanced matrix is that matrix if the number of cells in that matrix that are adjacent to that cell is less than the value written that cell.

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

Ans: It is useful to use list comprehension or any loop to create large matrices because without any looping we have to assign every single element to the matrix manually and it will create chaos in our code. We have to write so many lines of code containing variable assignments, which will eventually take a lot of time and space of the memory.