Question 1. Write a for loop to iterate through the list A = [1, 2, 3, 4, 5, 6]. Square each element of the list in one by one fashion and print them. After the end of the iteration, print - "The sequence has ended".

Output should look like this:

1

4

9

16

25

36

The sequence has ended

Question 2. If choice of user = 2, print the pattern - >

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

\_ \_

\_ \_ \_

\_ \_ \_ \_

\_ \_ \_ \_ \_

If choice of user = 1, print the pattern - >

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

If choice of the user = any\_other\_choice\_other\_than\_1\_and\_2, print the message - >

'Invalid Input'

Question 3. Create a tuple t\_1 = (1, 4, 9, 16, 25, 36). Square each element of the tuple using tuple comprehension and store the result in a variable known as t\_modified. Find element at index position 4 of the tuple t\_modified. Now slice the modified tuple in such a way that the sliced tuple includes only elements from index position 1 to 3 and store this sliced tuple in a variable known as t\_sliced.

Final output should be:

t\_1: (1, 4, 9, 16, 25, 36)

t\_modified: (1, 16, 81, 256, 625, 1296)

Element at index postiion 4 of t\_modified: 625

t\_sliced: (16, 81, 256)

Question 4. Show by raising a error how tuple are immutable and also define what exactly immutability is in your own words.

Question 5. Create a frozenset named frozen\_set\_1 containing the elements: 'A', 'B', 'C' and 'D' and combine it using union with a frozenset named frozen\_set\_2 containing elements 'A', 2, 'C' and 4. The final combined frozenset must be named frozenset\_union. Now find the common elements in frozen\_set\_1 and frozen\_set\_2 and store the result in a variable named frozenset\_common. Lastly, in a new forzenset named forzenset\_difference store the elements of frozen\_set\_1 which are not in frozen\_set\_2 and in a new frozenset named frozenset\_distinct store the elements which are unique to frozen\_set\_1 and frozen\_set\_2.

Final output:

frozen\_set\_1: frozenset({'C', 'A', 'B', 'D'})

frozen\_set\_2: frozenset({2, 'A', 'C', 4})

frozenset\_union: frozenset({2, 'A', 4, 'C', 'B', 'D'})

frozenset\_common: frozenset({'C', 'A'})

frozenset\_difference: frozenset({'D', 'B'})

frozenset\_distinct: frozenset({2, 'B', 4, 'D'})

Question 6. Write a python program to remove items in a list containing the character 'a' or 'A'. Use lambda function for it. For this program pass in as argument the list: list\_a = ["car", "place", "tree", "under", "grass", "price"] to the lambda function named remove\_items\_containing\_a\_or\_A.

Final output:

['tree', 'under', 'price']

Question 7: Create a custom exception class which can handle "IndexError" as well as "ValueError" such that it can display its own custom error message when we use index which is not valid in a list. Take list as list\_a = [1, 2, 3, 4, 5].

Final output type 1:

Enter the index = 10

The index 10 is incorrect and index should lie between -5 and 4.

Final output type 2:

Enter the index = abc

Use an Integer value as the input.