**Ex:** 6

**Date:** 13/10/2020

**Aim:**

To write and run the python program to solve the given questions and fill in the desired output

**Program:**

Predict the Output(1/4)

# Create a tuple, also called tuple packing.

numbers = 1, 2

print(numbers)

**(1, 2)**

# Create tuple with paranthesis.

numbers = (1, 2, 3)

print(numbers)

**(1, 2, 3)**

# Create an empty tuple.

numbers = ()

print(numbers)

**()**

# Create a tuple with one item. Note that the trailing comma is necessary

numbers = 1,

print(numbers)

**(1, )**

# Create a tuple with heterogenous items.

random\_tuple = "Hey", (1, 2), 1, ["you"]

print(random\_tuple)

**('Hey', (1, 2), 1, ['you'])**

# Create tuple with tuple() constructor.

numbers = tuple()

print(numbers)

**()**

numbers = tuple([1, 2]) # Takes any sequence as input

print(numbers)

**(1, 2)**

Predict the Output(2/4)

#### Methods on tuples #####

# Get length of list by using len() method.

numbers = 5, 8, 8

print(len(numbers))

**3**

# Get index of an element using the index() method.

numbers = 5, 8, 8

print(numbers.index(8))

**1**

# Count occurences of an item in a tuple.

numbers = 5, 8, 8

print(numbers.count(8))

**2**

eggs = ('hello', 42, 0.5)

eggs[0]

**hello**

eggs[1:3]

**(42, 0.5)**

len(eggs)

**3**

Predict the Output(3/4)

# Access elements of a tuple by indexing.

str\_tuple = "hey", "there!", "how", "are", "you?"

print(str\_tuple[0])

**hey**

print(str\_tuple[len(str\_tuple) - 1])

**you?**

print(str\_tuple[-1])

**you?**

# Slicing a tuple.

str\_tuple = "hey", "there!", "how", "are", "you?"

print(str\_tuple[2:])

**('how', 'are', 'you?')**

print(str\_tuple[:2])

**('hey', 'there!')**

print(str\_tuple[-3:])

**('how', 'are', 'you?')**

print(str\_tuple[:-3])

**('hey', 'there!')**

print(str\_tuple[1:4])

**('there!', 'how', 'are')**

# Get a copy of the tuple by slicing.

print(str\_tuple[:])

**('hey', 'there!', 'how', 'are', 'you?')**

Predict the Output(4/4)

# Concatenate tuples.

numbers = (1, 2)

strings = ("Hey", "there")

print(numbers + strings)

**(1, 2, 'Hey', 'there')**

# Looping through tuple using 'in'.

numbers = 1, 2

for number in numbers:

print(number)

**1 2**

# Check if element is present in tuple.

numbers = 1, 2

print(1 in numbers)

**True**

print(5 in numbers)

**False**

# Tuple packing.

# We are packing two items 1 and 2 into the tuple.

numbers = 1, 2

# Tuple sequence unpacking.

# Number of variables used has to be same as the number of items in the tuple.

# Unpacking the tuple and assigning its items to x and y.

x, y = numbers

# Note that this is also packing the args as a tuple which gets unpacked as the print method's arguments.

print(x, y)

**1 2**

**Result:**

Thus, running the python programs to obtain output for solving the given problem is done.