EXERCISE NO :6

DATE : 22-11-2020

AIM:

To write and run a Python program to fill in the desired output.

PROGRAM:

```
# Create a tuple, also called tuple packing.
numbers = 1, 2
print(numbers)
#output: (1, 2)

# Create tuple with paranthesis.
numbers = (1, 2, 3)
print(numbers)
#output: (1, 2, 3)

# Create an empty tuple.
numbers = ()
print(numbers)
#output: ()

# Create a tuple with one item. Note that the trailing comma is necessary
numbers = 1,
print(numbers)
#output: 1

# Create a tuple with heterogenous items.
random_tuple = "Hey", (1, 2), 1, ["you"]
print(random_tuple)
#output: ('Hey', (1, 2), 1, ['you'])
```

```
# Create tuple with tuple() constructor.
numbers = tuple()
print(numbers)
#output: ()

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

#### Methods on tuples #####
# Get length of list by using len() method.
numbers = 5, 8, 8
print(len(numbers))
#output: 3

# Get index of an element using the index() method.
numbers = 5, 8, 8
print(numbers.index(8))
#output: 1

# Count occurences of an item in a tuple.
numbers = 5, 8, 8
print(numbers.count(8))
#output: 2



eggs = ('hello', 42, 0.5)
eggs[0]
'hello'
```

```
#output: hello

eggs[1:3]

#output: (42, 0.5)


len(eggs)

#output: 3


# Access elements of a tuple by indexing.

str_tuple = "hey", "there!", "how", "are", "you?"

print(str_tuple[0])

#output: hey


print(str_tuple[len(str_tuple) - 1])

#output: you?


print(str_tuple[-1])

#output: you?


# Slicing a tuple.


str_tuple = "hey", "there!", "how", "are", "you?"

print(str_tuple[2:])

#output: ('how', 'are', 'you?')


print(str_tuple[:2])

#output: ('hey', 'there!')


print(str_tuple[-3:])

#output: ('how', 'are', 'you?')


print(str_tuple[:-3])
```

```
#output: ('hey', 'there!')

print(str_tuple[1:4])
#output: ('there!', 'how', 'are')

# Get a copy of the tuple by slicing.
print(str_tuple[:])
#output: ('hey', 'there!', 'how', 'are', 'you?')



# Concatenate tuples.
numbers = (1, 2)
strings = ("Hey", "there")
print(numbers + strings)
#output: (1, 2, "Hey", "there")

# Looping through tuple using 'in'.
numbers = 1, 2
for number in numbers:
  print(number)
 #output: 1 2

# Check if element is present in tuple.
numbers = 1, 2
print(1 in numbers)
#output: True

print(5 in numbers)
#output: 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)

#output: 1 2
```

RESULT:

Hence, the program is executed successfully.