Exercise No: 6
Date: 12/10/2020
AIM:
To predict the output for the given python program.
PROGRAM:
PREDICT THE OUTPUT:
# 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)
```

#### 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'

Hello

Eggs[1:3]

```
(42, 0.5)
```

Len(eggs)

3

# 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?')

# Concatenate tuples.

*Numbers* = (1, 2)

Strings = ("Hey", "there")

Print(numbers + strings)

(5, 8, 8, 'Hey', 'there')

(1, 2, "Hey", "there")

# Looping through tuple using 'in'.

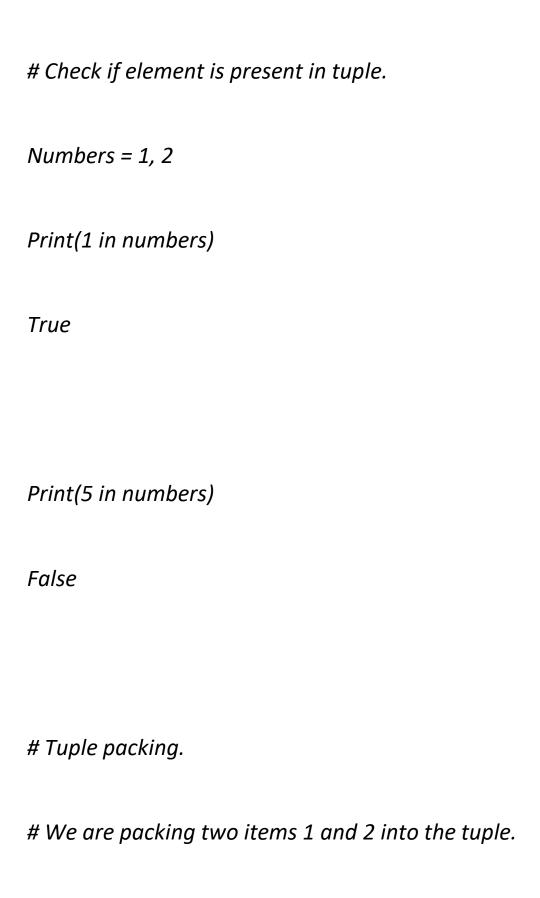
*Numbers* = 1, 2

For number in numbers:

*Print(number)* 

1,2

12



*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)

12

				•
•	1 N		_	•
•			N.	
_		v	•	-

http://103.53.53.18/mod/hvp/view.php?id=238

**RESULT:** 

The output for the given program is obtained.