Register no:**1518102067**

Exercise no:**6**

Date**:15/10/2020**

**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)    
# Create tuple with paranthesis.  
numbers = (1, 2, 3)  
print(numbers)   
# Create an empty tuple.  
numbers = ()  
print(numbers)   
# Create a tuple with one item. Note that the trailing comma is necessary  
numbers = 1,  
print(numbers)   
# Create a tuple with heterogenous items.  
random\_tuple = "Hey", (1, 2), 1, ["you"]  
print(random\_tuple)   
# Create tuple with tuple() constructor.  
numbers = tuple()  
print(numbers)   
numbers = tuple([1, 2]) #  Takes any sequence as input  
print(numbers) 

Check

Predict the Output

#### Methods on tuples #####  
# Get length of list by using len() method.  
numbers = 5, 8, 8  
print(len(numbers))   
# Get index of an element using the index() method.  
numbers = 5, 8, 8  
print(numbers.index(8))   
# Count occurences of an item in a tuple.  
numbers = 5, 8, 8  
print(numbers.count(8)) 

eggs = ('hello', 42, 0.5)   
eggs[0]   
eggs[1:3]   
len(eggs) 

Predict the Output

# Access elements of a tuple by indexing.  
str\_tuple = "hey", "there!", "how", "are", "you?"  
print(str\_tuple[0])   
print(str\_tuple[len(str\_tuple) - 1])   
print(str\_tuple[-1]) 

# Slicing a tuple.

str\_tuple = "hey", "there!", "how", "are", "you?"  
print(str\_tuple[2:])   
print(str\_tuple[:2])   
print(str\_tuple[-3:])   
print(str\_tuple[:-3])   
print(str\_tuple[1:4])   
# Get a copy of the tuple by slicing.  
print(str\_tuple[:]) 

Predict the Output

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

# Looping through tuple using 'in'.  
numbers = 1, 2  
for number in numbers:  
  print(number)   
    
# Check if element is present in tuple.  
numbers = 1, 2  
print(1 in numbers)   
print(5 in numbers) 

# 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) 

Check

**Link:**

[**http://103.53.53.18/mod/hvp/view.php?id=238**](http://103.53.53.18/mod/hvp/view.php?id=238)

**Result:** Thus the code has been executed successfully and the output was verified.