**Experiment No. 07**

**Program:Write a program to create an empty tuple.**

**Code:**

#Create an empty tuple x = ()

print(x)

#Create an empty tuple with tuple() function built-in Python tuple1 = tuple()

print(tuple1)

**Output:**

()

()

**Program:Write a Python program to create a tuple with different data types.**

**Code:**

#Create a tuple with different data types tuple2 = ("tuple", False, 3.2, 1) print(tuple2)

**Output:**

tuple2 = ("tuple", False, 3.2, 1)

**Program:**Write a Python program to get the 4th element and 4th element from last of a tuple.

**Code:**

#Get an item of the tuple

tuplex = ("U", "I", "T", 2, 0, 1, 8, "b", "a", "t", "c" , "h")

print(tuplex)

#Get item (4th element)of the tuple by index item = tuplex[3]

print(item)

#Get item (4th element from last)by index negative item1 = tuplex[-4]

print(item1)

**Output:**

('U', 'I', 'T', 2, 0, 1, 8, 'b', 'a', 't', 'c', 'h')

2

a

()

()

***Exercise:***

1. Use  inbuilt min and max functions to perform the task of getting the minimum and maximum value of in a list of tuples for a particular element position in a tuple.

Sample = [(2, 3), (4, 7), (8, 11), (3, 6)]

**Input:**

list\_of\_tuples=[(2,3),(4,7),(8,11),(3,6)]  
print("the original list is:",list\_of\_tuples)  
min\_max\_1=min(list\_of\_tuples)[0],max(list\_of\_tuples)[0]  
min\_max\_2=min(list\_of\_tuples)[1],max(list\_of\_tuples)[1]  
print("the min and max of index 1:",min\_max\_1)  
print("the min and max of index 2:",min\_max\_2)

**Output:**

the original list is: [(2, 3), (4, 7), (8, 11), (3, 6)]

the min and max of index 1: (2, 8)

the min and max of index 2: (3, 11)

2. A dartboard of radius 10 and the wall it is hanging on are represented using the two dimensional coordinate system, with the board’s center at coordinate (0; 0). Variables x and y store the x- and y-coordinate of a dart hit. Write an expression using variables x and y that evaluates to True if the dart hits (is within) the dartboard, and evaluate the expression for these dart coordinates:

(a) (0; 0)

(b) (10; 10) (c) (6; 6)

(d) (7; 8)

**Input:**

import math  
radius=10  
x=0  
y=0  
for i in range(4):  
 x1=int(input("enter value of x coordinate"))  
 y1=int(input("enter value of y coordinate"))  
 distance=math.sqrt((x-x1)\*\*2+(y-y1)\*\*2)  
 print("distance is:",distance)  
 if distance<radius:  
 print("hit")  
 else:  
 print("miss")

**Output:**

enter value of x coordinate0

enter value of y coordinate0

distance is: 0.0

hit

enter value of x coordinate10

enter value of y coordinate10

distance is: 14.142135623730951

miss

enter value of x coordinate6

enter value of y coordinate6

distance is: 8.48528137423857

hit

enter value of x coordinate7

enter value of y coordinate8

distance is: 10.63014581273465

miss

3. Write Python expressions corresponding to these statements:

(a)The number of characters in the word "anachronistically" is 1 more than the number of characters in the word "counterintuitive."

**a=(len("anachronistically")-len("counterintuitive")==1)**

(b)The word "misinterpretation" appears earlier in the dictionary than the word "misrep- resentation."

**b="misinterpretation"<"misrepresentation"**

(c)The letter "e" does not appear in the word "ﬂoccinaucinihilipiliﬁcation."

**c="e" not in "floccinaucinihilipilification**"

(d)The number of characters in the word "counterrevolution" is equal to the sum of the number of characters in words "counter" and "resolution."

**d=len("counterresolution")==len("counter")+len("resolution")**

4. Write a program in Python that holds an empty tuple and fill that tuple after taking user input for names of provinces of Pakistan n fill an empty tuple and print.

**Input:**

x=tuple()  
x1=list(x)  
for i in range(4):  
 y=input("province of Pakistan:")  
 x1.append(y)  
x=tuple(x1)  
print(x)

**Output:**

province of Pakistan:sindh

province of pakistan:punjab

province of Pakistan:balochistan

province of Pakistan:kpk

('sindh', 'punjab', 'balochistan', 'kpk')

5. Start by assigning to variables monthsL and monthsT a list and a tuple, respectively, both containing strings 'Jan', 'Feb', 'Mar', and 'May', in that order. Then attempt the following with both containers:

(a)Insert string 'Apr' between 'Mar' and 'May'. (b)Append string 'Jun'.

(c)Pop the container.

(d)Remove the second item in the container. (e)Reverse the order of items in the container.

**Note:** when attempting these on tuple monthsT you should expect errors.

**Input:**

#LIST

monthsL=["jan","feb","march","may"]

monthsL.insert(3,"april")

print(monthsL)

monthsL.append("june")

print(monthsL)

monthsL.pop()

print(monthsL)

monthsL.remove("feb")

print(monthsL)

monthsL.sort(reverse=True)

print(monthsL)

monthsL.sort()

print(monthsL)

print("\n")

#TUPLE

When doing it with tuple we except error so we first convert it into list then again convert it into tuple.

monthsT=("jan","feb","mar","may")

x=list(monthsT)

x.insert(3,"apr")

monthsT=tuple(x)

print(monthsT)

x=list(monthsT)

x.append("june")

monthsT=tuple(x)

print(monthsT)

x=list(monthsT)

x.pop()

monthsT=tuple(x)

print(monthsT)

x=list(monthsT)

x.remove("feb")

monthsT=tuple(x)

print(monthsT)

x=list(monthsT)

x.sort(reverse=True)

monthsT=tuple(x)

print(monthsT)

x=list(monthsT)

x.sort()

monthsT=tuple(x)

print(monthsT)

**Output:**

['jan', 'feb', 'march', 'april', 'may']

['jan', 'feb', 'march', 'april', 'may', 'june']

['jan', 'feb', 'march', 'april', 'may']

['jan', 'march', 'april', 'may']

['may', 'march', 'jan', 'april']

['april', 'jan', 'march', 'may']

('jan', 'feb', 'mar', 'apr', 'may')

('jan', 'feb', 'mar', 'apr', 'may', 'june')

('jan', 'feb', 'mar', 'apr', 'may')

('jan', 'mar', 'apr', 'may')

('may', 'mar', 'jan', 'apr')

('apr', 'jan', 'mar', 'may')