

## TYPE-C: TUPLES :CH-12

1) Write a Python program that creates a tuple storing first 9 terms of Fibonacci series.

sol:

```

lst = [0,1]
a = 0
b = 1
c = 0
for i in range(7):
    c = a + b
    a = b
    b = c
    lst.append(c)
tup = tuple(lst)
print("9 terms of Fibonacci series are:", tup)
```

output:

9 terms of Fibonacci series are: (0, 1, 1, 2, 3, 5, 8, 13, 21)

2)

a) Write a program that receives the index and returns the corresponding value.

sol:

```

tup=eval(input('enter the tuple:'))
num=int(input("enter the value of index:"))
a=tup[num]
print('at index',num,'the value is',a)
```

output:

enter the tuple:(10,20,30,40)  
 enter the value of index:2  
 at index 2 the value is 30

b) Write a program that receives a Fibonacci term and returns a number telling which term it is. For instance, if you pass 3, it returns 5, telling it is 5th term; for 8, it returns 7.

sol:

```

term = int(input ("Enter Fibonacci Term: "))
fib = (0,1)
while(fib[len(fib) - 1] < term):
    fib_len = len(fib)
    fib = fib + (fib[fib_len - 2] + fib[fib_len - 1],)
fib_len = len(fib)
if term == 0:
    print("0 is fibonacci term number 1")
elif term == 1:
    print("1 is fibonacci term number 2")
elif fib[fib_len - 1] == term:
    print(term, "is fibonacci term number", fib_len)
else:
    print("The term", term , "does not exist in fibonacci series")
```

output:

```
Enter Fibonacci Term: 7
The term 7 does not exist in fibonacci series
```

---

```
Enter Fibonacci Term: 5
5 is fibonacci term number 6
```

- 3) Write a program to input n numbers from the user. Store these numbers in a tuple. Print the maximum and minimum number from this tuple.

sol:

```
tup=()
n=int(input('enter total no.of numbers: '))
for i in range(n):
    num=int(input('enter the number:'))
    num=(num,)
    tup+=num
print('the tuple is: ',tup)
mx=max(tup)
mn=min(tup)
print('the maximum value is:',mx)
print('the minimum value is:',mn)
```

output:

```
enter total no.of numbers: 5
enter the number:10
enter the number:3
enter the number:5
enter the number:6
enter the number:7
the tuple is: (10, 3, 5, 6, 7)
the maximum value is: 10
the minimum value is: 3
```

- 4) Write a program to create a nested tuple to store roll number, name and marks of students.

sol:

```
list=[]
list1=[]
n=int(input('enter total no.of numbers: '))
for i in range(n):
    roll=int(input('enter the roll number:'))
    mark=int(input('enter the mark: '))
    name=input('enter the name: ')
    list.append(roll)
    list.append(mark)
    list.append(name)
    tuple2=tuple(list)
    list1.append(tuple2)
    tup=tuple(list1)
    list=[]
print(tup)
```

output:

```

enter total no.of numbers: 2
enter the roll number:16
enter the mark: 100
enter the name: shyam
enter the roll number:22
enter the mark: 83
enter the name: pranav
((16, 100, 'shyam'), (22, 83, 'pranav'))

```

- 5) Write a program that interactively creates a nested tuple to store the marks in three subjects for five students, i.e., tuple will look somewhat like :
- ```
marks( (45, 45, 40), (35, 40, 38), (36, 30, 38), (25, 27, 20), (10, 15, 20) )
```

sol:

```

list=[]
list1=[]
n=int(input('enter total no.of numbers: '))
for i in range(n):
    mark1=int(input('enter the mark1: '))
    mark2=int(input('enter the mark2: '))
    mark3=int(input('enter the mark3: '))
    list.append(mark1)
    list.append(mark2)
    list.append(mark3)
    tuple2=tuple(list)
    list1.append(tuple2)
    tup=tuple(list1)
    list=[]
print(tup)

```

output:

```

enter total no.of numbers: 2
enter the mark1:100
enter the mark2: 73
enter the mark3: 83
enter the mark1:87
enter the mark2: 92
enter the mark3: 87
((100, 73, 83), (87, 92, 87))

```

- 6) Write a program that interactively creates a nested tuple to store the marks in three subjects for five students and also add a function that computes total marks and average marks obtained by each student.

Tuple will look somewhat like :

```
marks( (45, 45, 40), (35, 40, 38),(36, 30, 38), (25, 27, 20), (10, 15, 20) )
```

sol:

```

tuple1=()
list=[]
for i in range(1,6):
    m1=int(input('enter the mark in subject 1 : '))

```

```

m2=int(input('enter the mark in subject 2 : '))
m3=int(input('enter the mark in subject 3 : '))
list.append((m1,m2,m3))
tuple1=tuple(list)
tuple2=(m1,m2,m3)
print('the sum of marks of student',i,'is: ',sum(tuple2))
print('the average of marks of student',i,'is: ',sum(tuple2)/3)
print('the nested tuple is: ',tuple1)

```

output:

```

enter the mark in subject 1 : 65
enter the mark in subject 2 : 74
enter the mark in subject 3 : 35
the sum of marks of student 1 is: 174
the average of marks of student 1 is: 58.0
enter the mark in subject 1 : 57
enter the mark in subject 2 : 54
enter the mark in subject 3 : 45
the sum of marks of student 2 is: 156
the average of marks of student 2 is: 52.0
enter the mark in subject 1 : 34
enter the mark in subject 2 : 65
enter the mark in subject 3 : 34
the sum of marks of student 3 is: 133
the average of marks of student 3 is: 44.33333333333336
enter the mark in subject 1 : 34
enter the mark in subject 2 : 65
enter the mark in subject 3 : 34
the sum of marks of student 4 is: 133
the average of marks of student 4 is: 44.33333333333336
enter the mark in subject 1 : 75
enter the mark in subject 2 : 34
enter the mark in subject 3 : 54
the sum of marks of student 5 is: 163
the average of marks of student 5 is: 54.33333333333336
the nested tuple is: ((65, 74, 35), (57, 54, 45), (34, 65, 34), (34, 65, 34), (75, 34, 54))

```

- 7) Write a program that inputs two tuples and creates a third, that contains all elements of the first followed by all elements of the second.

sol:

```

tuple1=eval(input('enter the tuple 1: '))
tuple2=eval(input('enter the tuple 2: '))
tuple3=tuple1+tuple2
print('the tuple is: ',tuple3)

```

output:

```

enter the tuple 1: (25,64,23)
enter the tuple 2: (67,98,45)
the tuple is: (25, 64, 23, 67, 98, 45)

```

- 8) Write a program as per following specification :

**"Return the length of the shortest string in the tuple of strings str\_tuple.  
Precondition: the tuple will contain at least one element."**

sol:

```
tuple1=eval(input('enter the tuple contains string: '))
list=[]
for i in tuple1:
    list.append(len(i))
a=min(list)
for i in tuple1:
    if len(i)==a:
        print('the smallest word is:',i)
```

output:

```
enter the tuple contains string: ('hello','python','world')
the smallest word is: hello
the smallest word is: world
```

9)

a) Create a tuple containing the squares of the integers 1 through 50 using a for loop.

sol:

```
tuple1=()
list=[]
for i in range(1,51):
    a=i**2
    list.append(a)
tuple1+=tuple(list)
list=[]
print('the tuple is: ',tuple1)
```

output:

```
the tuple is: (1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400,
441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444,
1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304, 2401, 2500)
```

b) Create a tuple ('a', 'bb', 'ccc', 'dddd', ...) that ends with 26 copies of the letter z using a for loop.

sol:

```
tuple1=()
list=[]
for i in range(1,27):
    a=i*chr(96+i)
    list.append(a)
tuple1+=tuple(list)
list=[]
print('the tuple is: ',tuple1)
```

output:

```
the tuple is: ('a', 'bb', 'ccc', 'dddd', 'eeee', 'fffff', 'gggggg', 'hhhhhhh', 'iiiiiii', 'jjjjjjjj',
'kkkkkkkkkk', 'llllllllll', 'mmmmmmmmmmmmmm', 'nnnnnnnnnnnnn', 'ooooooooooooooo',
'pppppppppppppppp', 'qqqqqqqqqqqqqqqqq', 'rrrrrrrrrrrrrrr', 'ssssssssssssssss',
'tttttttttttttttt', 'uuuuuuuuuuuuuuuuuuu', 'vvvvvvvvvvvvvvv',
```

```
'wwwwwwwwwwwwwwwwwwww', 'xxxxxxxxxxxxxxxxxxxxxx', 'yyyyyyyyyyyyyyyyyyyyy',
'zzzzzzzzzzzzzzzzzzzzz')
```

- 10) Given a tuple pairs = ((2, 5), (4, 2), (9, 8), (12, 10)), count the number of pairs (a, b) such that both a and b are even.

sol:

```
tuple1=((2,5),(4,2),(9,8),(12,10))
count=0
for i in tuple1:
    a,b=i
    if a%2==0 and b%2==0:
        print(i,'has two elementthat are even')
        count+=1
print('the count of pairs that has both the element as even numbers is:',count)
```

output:

```
(4, 2) has two elementthat are even
(12, 10) has two elementthat are even
the count of pairs that has both the element as even numbers is: 2
```

- 11) Write a program that inputs two tuples seq\_a and seq\_b and prints True if every element in seq\_a is also an element of seq\_b, else prints False.

sol:

```
seq_a=eval(input('ENTER THE TUPLE 1:'))
seq_b=eval(input('ENTER THE TUPLE 2:'))
if seq_a==seq_b:
    print('bothe the tuples are equal')
else:
    print('both the tuples are unequal')
```

output:

```
ENTER THE TUPLE 1:(1,2,3,4)
ENTER THE TUPLE 2:(1,2,3,4)
bothe the tuples are equal
-----
ENTER THE TUPLE 1:(1,2,3,4)
ENTER THE TUPLE 2:(1,2,3,5)
both the tuples are unequal
```

- 12) Computing Mean. Computing the mean of values stored in a tuple is relatively simple. The mean is the sum of the values divided by the number of values in the tuple. That is,

```
x
-
=
Σ
x
N
;
Σ
x
=
```

```

the sum of
x
,
N
=
number of elements
x
-
=
N
Σx

;
Σx=the sum of x,
N=number of elements

```

Write a program that calculates and displays the mean of a tuple with numeric elements.

sol:

```

t=eval(input('enter the tuple with numeric value:'))
count=0
for i in t:
    if str(i).isdigit():
        count+=1
if count==len(t):
    a=sum(t)
    print('the mean is: ',a/len(t))
else:
    print('the tuple should contain numeric value')

```

output:

```

enter the tuple with numeric value:(9,8,4,6,3)
the mean is: 6.0
-----
enter the tuple with numeric value:('hi','hello')
the tuple should contain numeric value

```

13) Write a program to check the mode of a tuple is actually an element with maximum occurrences.

sol:

```

import statistics as st
tup=eval(input('enter the tuple:'))
a=st.mode(tup)
print('mode is:',a)

```

output:

```

enter the tuple:(1,1,3,4,5,5,3,6,4,5)
mode is: 5

```

14) Write a program to calculate the average of a tuple's element by calculating its sum and dividing it with the count of the elements. Then compare it with the mean obtained using mean() of statistics module.

sol:

```

import statistics as st
tup=eval(input('enter the tuple:'))
a=st.mean(tup)
print('the mean by mean of method is:',a)
b=sum(tup)
c=len(tup)
mean=b/c
print('the mean by sum/len of method is:',mean)
if mean==a:
    print('the mean and average are same')
else:
    print('the mean and average are not same')

```

output:

```

enter the tuple:(67,43,72,98,81)
the mean by mean of method is: 72.2
the mean by sum/len of method is: 72.2
the mean and average are same

```

- 15) Mean of means. Given a nested tuple  $tup1 = ((1, 2), (3, 4.15, 5.15), (7, 8, 12, 15))$ . Write a program that displays the means of individual elements of tuple  $tup1$  and then displays the mean of these-computed means. That is for above tuple, it- should display as :

Mean element 1 : 1.5 ;  
 Mean element 2 : 4.1 ;  
 Mean element 3 : 10.5 ;  
 Mean of means 5.366666

sol:

```

import statistics as st
tup=((1,2),(3,4.15,5.15),(7,8,12,15))
le=len(tup)
for i in range(le):
    a=st.mean(tup[i])
    print('the mean of element',i,'is: ',a)

```

output:

```

the mean of element 0 is: 1.5
the mean of element 1 is: 4.1000000000000005
the mean of element 2 is: 10.5

```