

In []:

```

1 Create a tuple with 5 numbers and then add 2 more elements at the end
2 Add 3 elements which are already present in the tuple
3 remove the duplicates out of the whole list
4 get the memory location of this tuple m1
5 remove last element from this tuple and call it t2
6 get the memory location of t2
7 compare this with m1
8 iterate over the tuple and find the cube root of all the numbers in the tuple
9 take a list and convert it into a tuple and then into a set and
10 create another set from this set such that each element is equal to the cube of the
11 and create a dictionary with list values as keys and cubes values from the last set
12 create a tuple with only one element and create another tuple with 3 elements and call it t3
13 take a tuple and create a dictionary such that the elements in the tuples as keys and values as values
14 Take a tuple of 6 elements and create 3 tuples such that elements are consecutive elements of the tuple
15 take a tuple and find the harmonic mean of the min, max and len of the tuple
16 Take a tuple of 2 elements and assign it 2 variables and swap them
17 take a tuple of 3 elements and convert it into a string
18 take a tuple of 4 elements and find the sum of the indexes of 2nd and 3rd element
19 take a tuple of 5 elements with duplicates and find the count of 1 duplicated value
20 Take 2 tuples of equal lengths but of different elements and then compare whether they are equal or not
21 Take a list of 4 elements and convert into a tuple and a set
22 now find the memories of all the three and compare which has the highest

```

In [2]:

```

1 #Create a tuple with 5 numbers and then add 2 more elements at the end
2 this_tuple=(1,2,3,4,5)
3 tuple_1=(6,7)
4 this_tuple += tuple_1
5 print(this_tuple)

```

(1, 2, 3, 4, 5, 6, 7)

In [12]:

```

1 #Add 3 elements which are already present in the tuple
2 a=list(this_tuple)
3 a.append(1)
4 print(a)
5 b=(1,2,3)
6 this_tuple += b
7 print(this_tuple)

```

```

[1, 2, 3, 4, 5, 6, 7, 2, 3, 2, 3, 1, 2, 3, 1, 2, 3, 1]
(1, 2, 3, 4, 5, 6, 7, 2, 3, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3)

```

In [17]:

```

1 #remove the duplicates out of the whole list
2 c=tuple(set(this_tuple))
3 print(c)

```

(1, 2, 3, 4, 5, 6, 7)

In [21]:

```
1 #get the memory location of this tuple m1
2 m1=id(c)
3 print(m1)
```

1468095609632

In [22]:

```
1 #remove last element from this tuple and call it t2
2 e=list(c)
3 e.remove(7)
4 print(e)
```

[1, 2, 3, 4, 5, 6]

In [23]:

```
1 #get the memory location of t2
2 m2=id(e)
3 print(m2)
```

1468142553408

In [24]:

```
1 #compare this with m1
2 print(m1==m2)
```

False

In [14]:

```
1 #iterate over the tuple and find the cube root of all the numbers in the tuple
2 this_tuple=(1,2,3,4,5,6)
3
4 res=[]
5 for x in this_tuple:
6
7     res.append(x**3)
8
9 y=tuple(res)
10 print(y)
11
12
13
14
```

```
[1, 8, 27, 64, 125, 216]
(1, 8, 27, 64, 125, 216)
```

In [10]:

```
1 #take alist and convert it into a tuple and then into a set and
2 a_list=[1,2,3,4,5,6,7]
3 b_tuple=tuple(a_list)
4 print(b_tuple)
5 c_set=set(b_tuple)
6 print(c_set)
```

```
(1, 2, 3, 4, 5, 6, 7)
{1, 2, 3, 4, 5, 6, 7}
```

In [15]:

```
1 #create another set from this set such that each element is equal to the cube of the
2 res=[]
3 for x in c_set:
4
5     res.append(x**3)
6
7 y=set(res)
8 print(y)
```

```
{64, 1, 8, 343, 216, 27, 125}
```

In [17]:

```
1 #create a dictionary with list values as keys and cubes values from the last set as
2 a_list=[1,2,3,4,5,6,7]
3 a_dict=dict(zip(a_list,y))
4 print(a_dict)
```

```
{1: 64, 2: 1, 3: 8, 4: 343, 5: 216, 6: 27, 7: 125}
```

In [20]:

```
1 #create a tuple with only one element and create another tuple with 3 elements and c
2 b_tuple=(1,)
3 c_tuple=(4,2,3)
4 b_tuple += c_tuple
5 print(b_tuple)
```

```
(1, 4, 2, 3)
```

In [21]:

```
1 #take a tuple and create a dictionary such that the elements in the tuples as keys a
2 a_tuple=(2,3,4,5,6,7)
3 res=[]
4 for x in a_tuple:
5     res.append(x**2)
6 y=tuple(res)
7 z=dict(zip(a_tuple,y))
8 print(z)
9
10
```

```
{2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49}
```

In [29]:

```
1 #Take a tuple of 6 eleemnts and create 3 tuples such that elements are consecutive e
2 a_tuple=(1,2,3,4,5,6,7)
3 s=slice(0,3)
4 print(a_tuple[s])
5 s=slice(3,5)
6 print(a_tuple[s])
7 s=slice(5,7)
8 print(a_tuple[s])
```

```
(1, 2, 3)
(4, 5)
(6, 7)
```

In [33]:

```
1 #take a tuple and find the harmonic mean of the min, max and len of the tuple
2 length=len(a_tuple)
3 minimum=min(a_tuple)
4 maximum=max(a_tuple)
5 harmonic_mean=2*length*minimum*maximum/(length+minimum+maximum)
6 print(harmonic_mean)
```

6.533333333333333

In [35]:

```
1 #Take a tuple of 2 elements and assign it 2 variables and swap them
2 a=(9,0)
3 b=(8,7)
4 print(a,b)
5 a,b=b,a
6 print(a,b)
```

```
(9, 0) (8, 7)
(8, 7) (9, 0)
```

In [37]:

```
1 #take a tuple of 3 elements and convert it into a string
2 tuple_1=("b","a","d","a","p","p")
3 s=''.join(tuple_1)
4 print(s)
```

badapp

In [38]:

```
1 #take a tuple of 4 elements and find the sum of the indexes of 2nd and 3rd element
2 tuple_2=(4,5,6,7)
3 c=tuple_2[1]+tuple_2[2]
4 print(c)
```

11

In [40]:

```
1 #take a tuple of 5 elements with duplicates and find the count of 1 duplicated value
2 tuple_2=(9,9,0,1,2,1)
3 s=tuple_2.count(9)
4 print(s)
```

2

In [42]:

```
1 #Take 2 tuples of equal lengths but of different elements and then compare whether t
2 tuple_3=(6,7,8)
3 tuple_4=(3,4,5)
4 a=len(tuple_3)
5 b=len(tuple_4)
6 print(a==b)
7 print(tuple_3==tuple_4)
```

True

False

In [44]:

```
1 #Take a list of 4 elements and convert into a tuple and a set
2 list1=[1,2,3,4]
3 tuple1=tuple(list1)
4 print(tuple1)
5 set1=set(list1)
6 print(set1)
```

(1, 2, 3, 4)

{1, 2, 3, 4}

In [49]:

```
1 #now find the memories of all the three and compare which has the highest
2 l=id(list1)
3 print(l)
4 s=id(set1)
5 print(s)
6 t=id(tuple1)
7 print(t)
8 list1=[l,s,t]
9 def max_function(list1):
10     return max(list1)
11 max_function(list1)
```

2136353734976

2136354349120

2136353985296

Out[49]:

2136354349120

In []:

1	
---	--