CSE110 Lab Assignment 5

This Assignment is to help you develop your concept of Tuples and Dictionaries in Python.

Write your name, student ID and CSE110 theory section number below:

In [ ]:

```
#STUDENT NAME: M. ASADUZ ZAMAN
#STUDENT ID: 21101050
#CSE110 THEORY SECTION: 10
```

Write the code in Python to do the following tasks:

[MUST MAINTAIN VARIABLE NAMING CONVENTIONS FOR ALL THE TASKS]

Part 1: Tuple

Task 1

Assume, you have been given a tuple.

a_tuple = ("The Institute",\ ("Best Mystery & Thriller", "The Silent Patient", 68821),\ 75717,\ [1, 2, 3, 400, 5, 6, 7],\ ("Best Fiction", "The Testaments", 98291)\ )

Write **one line** of Python code to access and print the value 400.

===

Output: 400

===

In [1]:

```
a_tuple = ("The Institute",("Best Mystery & Thriller", "The Silent Patient", 68821),75717,[1, 2, 3, 400, 5, 6, 7],("Best Fiction", "The Testaments", 98291))
print(a_tuple[3][3])
```

```
400
```

Task 2

Assume, you have been given a tuple. Write a Python program that creates a **new tuple** excluding the first and last two elements of the given tuple and prints the new tuple.

Hint: You may use tuple slicing.

==

Sample Input 1:\ (10, 20, 24, 25, 26, 35, 70)

Sample Output 1:\ (24, 25, 26)

==

Sample Input 2:\ (-10, 20, 30, 40)

Sample Output 2:\ ()

==

Sample Input 3:\ (-10, 20, 25, 30, 40)

Sample Output 3:\ (25,)

==

In [2]:

```python
x=input("Enter Elements").split(",")
for i in range(len(x)):
    x[i]=int(x[i])
x=tuple(x)
lenght=len(x)
if lenght<4:
    print("Not Possible")
elif lenght==4:
    print()
else:
    print(x[2:-2])
```

```
Enter Elements10, 20, 24, 25, 26, 35, 70
(24, 25, 26)
```

Task 3

Assume, you have been given a tuple.

book_info = (\ ("Best Mystery & Thriller","The Silent Patient",68,821),\ ("Best Horror","The Institute",75,717),\ ("Best History & Biography","The five",31,783),\ ("Best Fiction","The Testaments",98,291)\ )

Write a Python program that prints the size of the tuple and all its elements as shown below.

===

Output:\ Size of the tuple is: 4\ ('Best Mystery & Thriller', 'The Silent Patient', 68, 821)\ ('Best Horror', 'The Institute', 75, 717)\ ('Best History & Biography', 'The five', 31, 783)\ ('Best Fiction', 'The Testaments', 98, 291)

===

In [3]:

```
book_info = (
("Best Mystery & Thriller","The Silent Patient",68,821),
("Best Horror","The Institute",75,717),
("Best History & Biography","The five",31,783 ),
("Best Fiction","The Testaments",98,291)
)
size=len(book_info)
print("Size of the tuple is:",size)
for i in book_info:
    print(i)
```

```
Size of the tuple is: 4
('Best Mystery & Thriller', 'The Silent Patient', 68, 821)
('Best Horror', 'The Institute', 75, 717)
('Best History & Biography', 'The five', 31, 783)
('Best Fiction', 'The Testaments', 98, 291)
```

Task 4

Assume, you have been given a tuple with details about books that won the Good Reads Choice Awards.

book_info = (\ ("Best Mystery & Thriller","The Silent Patient",68821),\ ("Best Horror","The Institute",75717),\ ("Best History & Biography","The five",31783),\ ("Best Fiction","The Testaments",98291)\ )

Write a Python program that prints the award category, the book name, and its total votes earned as shown below.

[Must use Tuple unpacking for printing and need to handle the quotation marks as a part of the output]

===

Output:\ The Silent Patient won the 'Best Mystery & Thriller' category with 68821 votes\ The Institute won the 'Best Horror' category with 75717 votes\ The five won the 'Best History & Biography' category with 31783 votes\ The Testaments won the 'Best Fiction' category with 98291 votes

===

In [1]:

```
book_info = (
("Best Mystery & Thriller","The Silent Patient",68821),
("Best Horror","The Institute",75717),
("Best History & Biography","The five",31783 ),
("Best Fiction","The Testaments",98291)
)
for i in book_info:
    category,name,votes=i
    print(name,"won the '" + category + "'category with",votes,"votes")
```

```
The Silent Patient won the 'Best Mystery & Thriller'category with 68821 votes
The Institute won the 'Best Horror'category with 75717 votes
The five won the 'Best History & Biography'category with 31783 votes
The Testaments won the 'Best Fiction'category with 98291 votes
```

Task 5

Write a python program that takes an input from the user and finds the number of times that the input is present in a given tuple.

===

Example 1:

Given tuple: (10, 8, 5, 2, 10, 15, 10, 8, 5, 8, 8, 2)

Sample Input 1:

8

Sample Output 1:

8 appears 4 times in the tuple

===

Example 2

Given tuple: (10, 8, 5, 2, 10, 15, 10, 8, 5, 8, 8, 2)

Sample Input 2:

1

Sample Output 2:

1 appears 0 times in the tuple

===

In [6]:

```
x=input("Enter a tuple")
count=0
num=input("Enter a num")
for i in range(len(x)):
    if x[i]==num:
        count+=1
print(num,"appers",count,"times in the tuple")
```

```
Enter a tuple10, 8, 5, 2, 10, 15, 10, 8, 5, 8, 8, 2
Enter a num8
8 appers 4 times in the tuple
```

Task 6

Write a Python program to reverse a given tuple.

[You are not allowed to use tuple slicing]

==

Note: Unlike lists, tuples are immutable. So, in order to reverse a tuple, we may need to convert it into a list first, then modify the list, and finally convert it back to a tuple.

==

Example 1:

Given tuple: ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h')

Output:

('h', 'g', 'f', 'e', 'd', 'c', 'b', 'a')

==

Example 2:

Given tuple: (10, 20, 30, 40, 50, 60)

Output:

(60, 50, 40, 30, 20, 10)

==

In [7]:

```
x=(input("Enter a tuple")).split(',')
y=[]
for i in  range(len(x)-1,-1,-1):
    value=x[i]
    if ord(value[0])>=48 and ord(value[0])<=57:
        y.append(int(value))
    else:
        y.append(value)
t=tuple(y)
print(t)
```

```
Enter a tuplea,a,e,w,r,t,f,s
('s', 'f', 't', 'r', 'w', 'e', 'a', 'a')
```

Part 2: Dictionary

Task 7

Suppose you are given two dictionaries.\ Now create a new dictionary "marks", merging the two dictionaries, **so that the original two dictionaries remain unchanged.**

Note: You can use dictionary functions.

==

Given:

{'Harry':15, 'Draco':8, 'Nevil':19}

{'Ginie':18, 'Luna': 14}

Output:

{'Harry': 15, 'Draco': 8, 'Nevil': 19, 'Ginie': 18, 'Luna': 14}

==

Given:

{'A':90, 'B': 0}

{'C':50}

Output:

{'A': 90, 'B': 0, 'C': 50}

==

In [13]:

```
a={'Harry':15, 'Draco':8, 'Nevil':19}
a['Ginie']=18
a['Luna']= 14
print(a)
```

```
{'Harry': 15, 'Draco': 8, 'Nevil': 19, 'Ginie': 18, 'Luna': 14}
```

Task 8

Write a Python program that takes a dictionary as an input from the user and then prints the average of all the values in the dictionary.

[You are not allowed to use len() and sum()]

===

Hint (1): For taking dictionary input

Approach(1): For taking dictionary as an input from the user, you may take the whole dictionary as a string using the input() function. Then you can use the split(), strip() functions and conditions to get the keys and values from the string. Finally, you can make the dictionary using the obtained data.

Approach(2): If the first approach seems too difficult you can create an empty dictionary and then just run a simple loop. For each iteration ask the user for a key and a value using the input() function and keep updating the dictionary with the key and value.

Hint (2): After you have a dictionary, you can use dictionary functions to get all the values from it, run loop to calculate the sum and the total number of values in the dictionary in order to find out the average.

===

Sample Input 1:

{'Jon': 100, 'Dan':200, 'Rob':300}

Sample Output 1:

Average is 200.

===

Sample Input 2:

{'Jon': 100, 'Dan':200, 'Rob':30, 'Ned':110}

Sample Output 2:

Average is 110.

===

In [9]:

```
x=input("Write a Dictionary : ")[1:-1].split(",")
d={}
for  i in x:
    k,v=i.split(":")
    k=k.strip()[1:-1]
    v=int(v)
    d[k]=v
c=0
s=0
for v in d.values():
    s+=v
    c+=1
print(s//c)
```

```
Write a Dictionary : {'Jon': 100, 'Dan':200, 'Rob':300}
200
```

Task 9

Suppose there is a dictionary named exam_marks as given below.

exam_marks = {'Cierra Vega': 175, 'Alden Cantrell': 200, 'Kierra Gentry': 165, 'Pierre Cox': 190}

Write a Python program that takes an input from the user and creates a new dictionary with only those elements from 'exam_marks' whose keys have values higher than the user input (inclusive).

==

Sample Input 1:

170

Sample Output 1:

{'Cierra Vega': 175, 'Alden Cantrell': 200, 'Pierre Cox': 190}

==

Sample Input 2:

190

Sample Output 2:

{'Alden Cantrell': 200, 'Pierre Cox': 190}

==

In [8]:

```
exam_marks = {'Cierra Vega': 175, 'Alden Cantrell': 200, 'Kierra Gentry': 165,
'Pierre Cox': 190}
marks=int(input("Enter the mark: "))
a={}
for k,v in exam_marks.items():
    if v>=marks:
        a[k]=v
print(a)
```

```
Enter the mark: 170
{'Cierra Vega': 175, 'Alden Cantrell': 200, 'Pierre Cox': 190}
```

Task 10

Write a Python program that finds the largest value with its key from a given dictionary. [without using max() function]

Note: You do not need to take the dictionaries as an input from the user but your code should work for any given dictionary. Also, you need to handle the quotation marks as a part of the output.

Hint: Think of membership operators (in and not in). You can use dictionary functions to get the values.

===

Sample 1:

Given:

{'sci fi': 12, 'mystery': 15, 'horror': 8, 'mythology': 10, 'young_adult': 4, 'adventure':14}

Output:

The highest selling book genre is 'mystery' and the number of books sold are 15.

===

Sample 2:

Given:

{'sci fi': 5, 'mystery': 3, 'horror': 14, 'young_adult': 2, 'adventure':9}

Output:

The highest selling book genre is 'horror' and the number of books sold are 14.

===

In [1]:

```
x=input("enter a dictionary : ")[1:-1].split(",")
d={}
for str_elem in x :
    k,v=str_elem.split(":")
    k=k.strip()[1:-1]
    v=int(v)
    d[k]=v
max_key=None
max_value=-1
for k,v in d.items():
    if v>max_value:
        max_value=v
        max_key=k
print(f"The highest selling book grnre is '{max_key} and the number of books sold are {max_value}.")
```

```
enter a dictionary : {'sci fi': 12, 'mystery': 15, 'horror': 8, 'mythology':
10, 'young_adult': 4, 'adventure':14}
The highest selling book grnre is 'mystery and the number of books sold are 1
5.
```

Task 11

Write a Python program that takes a String as an input from the user and counts the frequency of each character using a dictionary. For solving this problem, you may use each character as a key and its frequency as values. [You are not allowed to use the count() function]

Hint: You can create a new dictionary to store the frequencies. You may ignore case for simplicity (i.e. may consider P and p to be the same).

===

Sample Input:\ "Python programming is fun"

Sample Output:\ {'p': 2, 'y': 1, 't': 1, 'h': 1, 'o': 2, 'n': 3, 'r': 2, 'g': 2, 'a': 1, 'm': 2, 'i': 2, 's': 1, 'f': 1, 'u': 1}

===

In [3]:

```
x=(input("Enter a string : ").lower())
y={}
for i in x :
    if i !=" ":
        if i not in y:
            y[i]=1
        else:
            y[i]+=1
print(y)
```

```
Enter a string : Python programming is fun
{'p': 2, 'y': 1, 't': 1, 'h': 1, 'o': 2, 'n': 3, 'r': 2, 'g': 2, 'a': 1, 'm':
2, 'i': 2, 's': 1, 'f': 1, 'u': 1}
```

Task 12

Suppose you are given the following dictionary where the values are lists.

dict = {'A': [1, 2, 3], 'b': ['1', '2'], "c": [4, 5, 6, 7]}

Write a Python program that counts the total number of items in a dictionary's values and prints it. [without using sum(), len(), count() functions]

Note: Make changes to the above dictionary and see if your code works properly for other dictionaries as well.

===

Output:

9

===

In [4]:
```
dict = {'A': [1, 2, 3], 'b': ['1', '2'], "c": [4, 5, 6, 7]}
c=0
for value in dict.values():
    for i in value:
        c+=1
print(c)
```

```
9
```

Task 13

Suppose you have been given the following list of tuples.

list = [("a", 1), ("b", 2), ("a", 3), ("b", 1), ("a", 2), ("c", 1)]

Write a Python program that converts this list of tuples into a dictionary and then prints the dictionary. [You are not allowed to use set]

Hint: Think of membership operators (in and not in).

===

Output:

{'a': [1, 3, 2], 'b': [2, 1], 'c': [1]}

===

In [5]:

```
list = [("a", 1), ("b", 2), ("a", 3), ("b", 1), ("a", 2), ("c", 1)]
dict= {}
for item in list:
    k=item[0]
    v=item[1]
    if k not in dict:
        dict[k]=[v]
    else:
        dict[k]+=[v]
print(dict)
```

```
{'a': [1, 3, 2], 'b': [2, 1], 'c': [1]}
```

Optional Tasks (14-18) [Ungraded]

Task 14

Given a list of tuples, your task is to multiply the elements of the tuple and return a list of multiplied elements as shown below.

===

Example 1:

Given:

[(2, 3), (4, 5), (6, 7), (2, 8)]

Output:

[6, 20, 42, 16]

===

Example 2:

Given:

[(11, 22), (33, 55), (55, 77), (11, 44)]

Output:

[242, 1815, 4235, 484]

===

In [ ]:

```
#to do
```

Task 15

Assume, you have been given a tuple as below.

a_tuple = ([1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12])

Write a Python program that asks the user for an input (can be any data type) and **replaces the last element of each of the inner lists with the user given value.**

===

Sample Input 1:

abc

Sample Output 1:

([1, 2, 'abc'], [4, 5, 'abc'], [7, 8, 'abc'], [10, 11, 'abc'])

===

Sample Input 1:

1000

Sample Output 1:

([1, 2, '1000'], [4, 5, '1000'], [7, 8, '1000'], [10, 11, '1000'])

===

In [ ]:
```
#todo
```

Task 16

Suppose there is a dictionary named my_dictionary.

my_dictionary = {'c1':'Red', 'c2':'Green', 'c3':None, 'd4':'Blue', 'a5':None}.

Now write a Python program to remove empty items from the dictionary. [Empty items means keys without any values (None)].

===

Output:

{'c1':'Red', 'c2':'Green', 'd4':'Blue'}

===

In [ ]:
```
#to do
```

Task 17

Suppose you are given a dictionary as shown below. Write a Python program that takes two inputs from the user representing the lower (inclusive) and upper (exclusive) of a range of values. Your task is to extract all the items from the dictionary whose values lie in the range given as input by the user.

dictionary = {'a' : 6, 'b' : 7, 'c' : 9, 'd' : 8, 'e' : 11, 'f' : 12, 'g' : 13}

==

Sample Input 1:

9, 12

Sample Output 1:

{'c': 9, 'e': 11}

Explanation: Keys with values within the range of 9 and 11 are extracted.

==

Sample Input 2:

14, 18

Sample Output 2 :

{}

Explanation: No values in range.

==

```
In [ ]: #to do
```

Task 18

Given a list of tuples, your task is to group the tuples based on the **second** element in the tuples as shown in the examples below. We can achieve this using **dictionary** by checking the second element in each tuple.

[You do not need to take tuple as input and can assume that it is given as below]

==

Example 1:

Given:

[(20, 80), (31, 80), (1, 22), (88, 11), (27, 11)]

Output:

{80: [(20, 80), (31, 80)],\ 11: [(88, 11), (27, 11)],\ 22: [(1, 22)]}

==

Example 2:

Given:

[(20, 'Sad'), (31, 'Sad'), (88, 'NotSad'), (27, 'NotSad')]

Output:

{'NotSad': [(88, 'NotSad'), (27, 'NotSad')],\ 'Sad: [(20, 'Sad'), (31, 'Sad')]}

==

```
In [ ]: #todo
```