2347227_Lab exercise 1

August 5, 2023

Joe Payyappilly-2347227-MCA B-Lab Exercise 1

1.Introduction Paragraph and frequency counting

```
[ ]: paragraph="My name is Joe Payyappilly. My register number is 2347227. I am a␣
      ↪MCA 1st semester student. My domain name is Physics."

     def total(paragraph, word):
         word_list = paragraph.lower().split()
         word_times = word_list.count(word.lower())
         return word_times

     myword="my"
     times = total(paragraph, myword)
     print(f"The word '{myword}' appears {times} times in the paragraph.")
```

```
The word 'my' appears 3 times in the paragraph.
```

2.Python program to display datatypes of specific elements in the paragraph:

```
[ ]: def identify_datatypes(paragraph):
         elements = paragraph.split()
         datatypes = {}
         for element in elements:
             if element.isdigit():
                 datatypes[element] = "int"
             elif element.replace('.', '', 1).isdigit():
                 datatypes[element] = "float"
             else:
                 datatypes[element] = "string"
         return datatypes

     paragraph = "My name is Joe Payyappilly. My register number is 2347227 . I am a␣
      ↪MCA 1st semester student. My domain name is Physics ."
     datatypes = identify_datatypes(paragraph)
     for element, datatype in datatypes.items():
         print(f"{element} - {datatype}")
```

```
My - string
name - string
```

```
is - string
Joe - string
Payyappilly. - string
register - string
number - string
2347227 - int
. - string
I - string
am - string
a - string
MCA - string
1st - string
semester - string
student. - string
domain - string
Physics - string
```

3.Python program to count the number of alphabets, numerics, and special symbols in the paragraph:

```
[ ]: def count_symbols(paragraph):
         alphabets = numerics = specials = 0
         for char in paragraph:
             if char.isalpha():
                 alphabets += 1
             elif char.isdigit():
                 numerics += 1
             else:
                 specials += 1
         return alphabets, numerics, specials

     paragraph = "My name is Joe Payyappilly. My register number is 2347227. I am a␣
       ↪MCA 1st semester student. My domain name is Physics."
     num_alphabets, num_numerics, num_specials = count_symbols(paragraph)
     print(f"Number of alphabets: {num_alphabets}")
     print(f"Number of numeric characters: {num_numerics}")
     print(f"Number of special symbols: {num_specials}")
```

```
Number of alphabets: 85
Number of numeric characters: 8
Number of special symbols: 25
```

4. Perform the functions pop(), clear(), discard() and del.

```
[ ]: phy_set={"Scientists",10,True,"equations",8.314,False}
     print(phy_set)
     popped_element = phy_set.pop()
     print(phy_set)
     print(popped_element)
```

```
phy_set.discard(10)
print(phy_set)
phy_set.clear()
print(phy_set)
del phy_set

"""
Insights:
1. The pop() function removes and returns an arbitrary element from the set. As␣
 ↪sets are unordered, we cannot predict which element will be removed.
2. The clear() function removes all elements from the set, leaving it empty.
3. The discard() function removes a specific element from the set if it exists.␣
 ↪If the element is not present, it does nothing and does not raise an error.
4. The del statement deletes the entire set, and trying to access the set after␣
 ↪deletion will raise an error.
"""
```

```
{False, True, 'equations', 8.314, 'Scientists', 10}
{True, 'equations', 8.314, 'Scientists', 10}
False
{True, 'equations', 8.314, 'Scientists'}
set()
```

```
[ ]: '\nInsights:\n1. The pop() function removes and returns an arbitrary element
     from the set. As sets are unordered, we cannot predict which element will be
     removed.\n2. The clear() function removes all elements from the set, leaving it
     empty.\n3. The discard() function removes a specific element from the set if it
     exists. If the element is not present, it does nothing and does not raise an
     error.\n4. The del statement deletes the entire set, and trying to access the
     set after deletion will raise an error.\n'
```

5.Updating set

```
[ ]: phy_set = {"Scientist", "Equations", "Theories", 2.314, True}
     print(phy_set)
     phy_set.update(["Experiments", 10, False])
     print(phy_set)
```

```
{True, 2.314, 'Theories', 'Scientist', 'Equations'}
{False, True, 2.314, 'Experiments', 'Theories', 10, 'Scientist', 'Equations'}
```

6.Sorting set

```
[ ]: phy_set = {"Scientist", "Equations", "Theories"}
     sorted_phy_set = sorted(phy_set, reverse=True)
     print(sorted_phy_set)
```

```
['Theories', 'Scientist', 'Equations']
```

7.Tuple Packing and unpacking

```
[ ]: tuple_phy = ("Equation", "Theory", "Scientist")
     (attr1, attr2, attr3) = tuple_phy
     print(attr1)
     print(attr2)
     print(attr3)
```

```
Equation
Theory
Scientist
```

8.Counting

```
[ ]: domain = ("p","h","y","s","i","c","s")
     count = domain.count('s')
     print(count)
```

```
2
```

9.Slicing and Indexing

```
[ ]: domain_name = "physics"
     slice1 = domain_name[0:3]
     print(slice1)
     slice2 = domain_name[2:5]
     print(slice2)
     slice3 = domain_name[:5]
     print(slice3)
     slice4 = domain_name[3:]
     print(slice4)
     slice5 = domain_name[:]
     print(slice5)
     slice6 = domain_name[-1]
     print(slice6)
     slice7 = domain_name[-3:]
     print(slice7)
     slice8 = domain_name[-6:-3]
     print(slice8)
     slice9 = domain_name[:-2]
     print(slice9)
     slice10 = domain_name[-5: -1]
     print(slice10)
```

```
phy
ysi
physi
sics
physics
s
ics
hys
```

```
physi
ysic
```