01

July 30, 2023

1. Demonstrate use of Python data structures.

a. Write a paragraph on your identified domain. Write a python program to find the frequency of the given word(your domain name) in the paragraph. to compute the number of characters, words and lines in the paragraph to arrange the letters of the given word(your domain name) in alphabetical order

[ ]:
```
domianDesc = "A Content Management System (CMS) is a powerful software␣
 ↪application designed to simplify the creation, management, and publication␣
 ↪of digital content on websites. It provides users, whether they are␣
 ↪individuals or organizations, with a user-friendly interface to effortlessly␣
 ↪handle various aspects of content, such as writing, editing, organizing, and␣
 ↪scheduling. CMS platforms enable multiple users to collaborate seamlessly,␣
 ↪facilitating efficient workflows for content creation and approval processes.
 ↪ With its diverse array of tools and templates."

print("Number of occurence of the word 'CMS' : ", domianDesc.count("CMS"))

countChar = 0
for i in domianDesc:
    if (i == ',' or i == ' ' or i == '.'):
        pass
    else:
        countChar += 1

print("Number of characters in the above description is : ", countChar) #number␣
 ↪of char
print("Number of words in the above description is : ",
      len(domianDesc.split(' ')))  # number of words
print("Number of lines in the above description is : ",
      len(domianDesc.split('.')))  # number of sentences
```

```
Number of occurence of the word 'CMS' :  2
Number of characters in the above description is :  466
Number of words in the above description is :  75
Number of lines in the above description is :  5
```

b. Write a python program to encrypt a given string(your domain name) using the following method: Encrypt Method: Add a number ‘n’(given by the user) to each alphabet in the given string to create the corresponding letter. Example: Input: bat Encrypt Method:value of n = 3 Output:edw

[ ]:
```
import string
domainName = "content management system"
n = int(input("Enter the add number"))
indexList = []
domainName.split(' ')
for i in domainName:
    if (string.ascii_lowercase.find(i) >= (26-n)):
        i = 'a'
    indexList.append(string.ascii_lowercase[(
        string.ascii_lowercase.find(i))+n])
print("Encrypt Method:value of n = ", n)
print("Encrypted string is : ", "".join(indexList))
```

```
Encrypt Method:value of n =  5
Encrypted string is :  htsyjsyerfsfljrjsyexfxyjr
```

2. Functions : Implement a function pay() that takes as input two arguments: an hourly wage and the number of hours an employee worked last week. Your function should compute and return the employee's pay. Any hours worked beyond 40 is overtime and should be paid 1.5 times the regular hourly wage. »> pay(10,35) 350 »>pay(10,45) 475.0

[ ]:
```
def pay(hourlyWage, WorkHour):
    if WorkHour > 40:
        return (hourlyWage*1.5)*WorkHour
    return hourlyWage*WorkHour

hourlyWage, workHour = [float(s) for s in
                        input("enter hourly wage and work hour of the employee␣
 ↪").split(' ')]
print("employee's pay : ", pay(hourlyWage, workHour))
```

```
employee's pay :  36.0
```

3. Tuple : Create a list of tuples that consists of two neumatic and one string For example houses for rent, the number of bedrooms and their prices, like so: [('main st.', 4, 4000), ('elm st.', 1, 1200), ('pine st.', 2, 1600)] Sort the list in the following ways:

a. In ascending order by first numeric value
b. In descending order by second numeric value
c. In alphabetical order of string value

[ ]:
```
houseList = [('main st.', 4, 4000), ('elm st.',
                                     1, 1200), ('pine st.', 2, 1600)]
streetName = []
numVal = []
rentList = []
```

```
for i in houseList:
    streetName.append(i[0])
    numVal.append(i[1])
    rentList.append(i[2])
print("alphabetical order of string value(Street Name) : ", sorted(streetName))
print("descending order by first numeric value : ", sorted(numVal))
print("descending order by second numeric value (Rent) :",
      sorted(rentList, reverse=True))
```

```
alphabetical order of string value(Street Name) :  ['elm st.', 'main st.', 'pine
st.']
descending order by first numeric value :  [1, 2, 4]
descending order by second numeric value (Rent) : [4000, 1600, 1200]
```

Using itemgetter

[ ]:
```
from operator import itemgetter
houseList = [('main st.', 4, 4000), ('elm st.',
                                     1, 1200), ('pine st.', 2, 1600)]
print("alphabetical order of string value(Street Name) : ",
      sorted(houseList, key=itemgetter(0)))
print("descending order by first numeric value : ",
      sorted(houseList, key=itemgetter(1)))
print("descending order by second numeric value (Rent) :",
      sorted(houseList, key=itemgetter(2), reverse=True))
```

```
alphabetical order of string value(Street Name) :  [('elm st.', 1, 1200), ('main
st.', 4, 4000), ('pine st.', 2, 1600)]
descending order by first numeric value :  [('elm st.', 1, 1200), ('pine st.',
2, 1600), ('main st.', 4, 4000)]
descending order by second numeric value (Rent) : [('main st.', 4, 4000), ('pine
st.', 2, 1600), ('elm st.', 1, 1200)]
```