PYTHON Lab_2 Assignment

Meghana D Duttargi

241059042

ME CYS, MSIS

String Operations in Python

1. Find the length of the string

```
#!/usr/bin/python

string="hello world"
print(len(string))
```

2. Slice the string as per your choice

```
#string slicing
string1 = "HEY"
s1 = slice(1)
print(string1[s1])
~
~
~
```

3. Concatenate two strings

```
#Concatenate two strings
str1 = "HELLO"
str2 = "HEY"
res = str1 + str2
print(res)
~
~
```

4. Convert in to lower case in to uppercase character

```
#Lowercase to uppercase
string2 = "hello"
s2 = string2.upper()
print(s2)
```

5. Convert upper case into lower case characters

```
#Lowercase to uppercase
string2 = "hello"
s2 = string2.upper()
print(s2)
#uppercase to lowercase
string3 = "HELLO"
print(string3.lower())
~
~
```

6. convert the character into Unicode (Ascii values)

```
#character to unicode (ASCII VAlue)
s3 = input("")
temp = ord(s3)
print(temp)
~
~
```

7. convert Unicode into character

```
#unicode to character
s4 = 97
temp1 = chr(s4)
print(temp1)
~
~
```

8. Check whether the given "substring" exists in the string

```
string4 = "Hello world"
substring = "world"
if substring in string4:
  print("Substring found!")
else:
  print("Substring not found.")
~
```

9. Replace the character 'k' with 'h'

```
#replace k with h
string5 = "kitchen"
new_string = string5.replace('k', 'h')
print(new_string)
```

10. Pad the string with "x" at the end

```
#padding with x at the end
string6 = "hello"
padded_string = string6.ljust(10, 'x')
print(padded_string)
```

11. remove leading and trailing whitespace or specified characters from the string

```
#removing leading and trailing whitespace
string7 = "    Hello World!    "
trimmed_string = string7.strip()
print(trimmed_string)
```

12. split the given string in to group of five characters

```
#split the string into group of 5 characters
string8 = "HelloWorldwelcome"
split_string = [string8[i:i+5] for i in range(0, len(string8), 5)]
print(split_string)
```

13. count total number of words

```
#count total number of words
string9 = "Hello, how are you doing today?"
words = string9.split()
word_count = len(words)
print(word_count)
```

14. Find the frequency of each characters in the string

```
#frequency of each characters in the string
from collections import Counter
string10 = "hello world"
frequency = Counter(string10)
print(frequency)
```

```
┌──(kali㉿kali)-[~]
└─$ python lab2.py
11
H
HELLOHEY
HELLO
hello
a
97
a
Substring found!
```

```
hitchen
helloxxxxx
Hello World!
['Hello', 'World', 'welco', 'me']
6
Counter({'l': 3, 'o': 2, 'h': 1, 'e': 1, ' ': 1, 'w': 1, 'r': 1, 'd': 1})
```

STDIN and File operators

15. get the file name from the user

16. check the file exist or not

```python
file_name = input("Please enter the file name: ")

print(f"The file name you entered is: {file_name}")

import os

file_name = input("Please enter the file name: ")

if os.path.exists(file_name):
    print("The file exists.")
else:
    print("The file does not exist.")
```

```
┌──(kali㉿kali)-[~]
└─$ python stdin_file_op.py
Please enter the file name: abc
The file name you entered is: abc
Please enter the file name: missionary
The file does not exist.
```

Looping and File handling

17. read the contents from the file

18. reverse the contents from the file

19. Write into the file

```python
with open(file_name, 'r') as file:
            content = file.read()
            content ="HELLO"
            print(content)
with open(fikle_name, "r") as file:
    content = file.read()
    reversed_content = content[::-1]
    print(reversed_content)  # Correctly indented


with open(file_name, 'a') as file:
    file.write("\nAppended text.")
    print("Text appendedto file.")
```

Math operations

20. convert Frequency in to percentage (continuation of 12th Question)

```python
import math
#20.Frequency into percentage
items = ['cryptology', 'linux', 'linux', 'networks','linux']

frequency = {}
for item in items:
    if item in frequency:
        frequency[item] += 1
    else:
        frequency[item] = 1

total_items = len(items)

percentage = {item: (count / total_items) * 100 for item, count in frequency.items()}

for item, perc in percentage.items():
    print(f"{item}: {perc:.2f}%\n")
```

21. Perform modular arithmetic operation

```python
#21.Modular arithmetic operation
a= 10
b= 2
modulus = a%b
print(f"modulus: \n", modulus)
```

22. Find the prime numbers

check the given number is prime or not

print the prime numbers with the given range

```python
#22.Prime Numbers operation
def is_prime(n):
    if n <= 1:
        return False
    if n == 2 or n == 3:
        return True
    if n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while i * i <= n:
        if n % i == 0 or n % (i + 2) == 0:
            return False
        i += 6
    return True

def print_primes_in_range(start, end):
    print(f"Prime numbers between {start} and {end} are:")
    for num in range(start, end + 1):
        if is_prime(num):
            print(num, end=" ")
    print()


number = 7
if is_prime(number):
    print(f"\n{number} is a prime number.")
else:
    print(f"{number} is not a prime number.\n")


start = 10
end = 100
print_primes_in_range(start, end)
```

23. Check the given two numbers are co prime or not

```python
#23.Co prime or not
import math

def are_coprime(a, b):
    return math.gcd(a, b) == 1


num1 = 2
num2 = 3

if are_coprime(num1, num2):
    print(f"\n{num1} and {num2} are co-prime.")
else:
    print(f"{num1} and {num2} are not co-prime.\n")
```

24. find the factors for the given number (can use python library)

```python
#24.factorisation
import sympy

def find_factors(n):
    # Using sympy to get the factors
    factors = sympy.divisors(n)
    return factors

number = 4
factors = find_factors(number)

print(f"\nFactors of {number} are: {factors}")
```

25. generate 10 random numbers

```python
#25.generate 10 random numbers
import random

random_numbers = [random.randint(1, 100) for _ in range(10)]

print("\n10 random integers:", random_numbers)
```

26. Explore : Miller-Rabin Test (pen paper method)

```python
#26.Miller-Rabin test

def miller_rabin(n, k):

    # Handle small numbers
    if n <= 1:
        return False
    if n <= 3:
        return True
    if n % 2 == 0 or n % 3 == 0:
        return False

    # Write n-1 as 2^s * d
    d = n - 1
    s = 0
    while d % 2 == 0:
        d //= 2
        s += 1

    # Perform the test k times
    for _ in range(k):
        a = random.randint(2, n - 2)
        x = pow(a, d, n)  # Compute a^d % n

        if x == 1 or x == n - 1:
            continue

        for _ in range(s - 1):
            x = pow(x, 2, n)
            if x == n - 1:
                break
        else:
            return False

    return True


number = 22
k = 5  # Number of iterations
if miller_rabin(number, k):
    print(f"\n{number} is probably prime.")
else:
    print(f"\n{number} is composite.")
```

```
cryptology: 20.00%

linux: 60.00%

networks: 20.00%

modulus:
 0

7 is a prime number.
Prime numbers between 10 and 100 are:
11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

2 and 3 are co-prime.

Factors of 4 are: [1, 2, 4]

10 random integers: [15, 22, 45, 16, 11, 80, 96, 19, 58, 73]

22 is composite.
```

“Don’t worry about failures, worry about the chances you miss when you don’t even try.”

– JACK CANFIELD