Python Assignment

1. Find the length of the string? my_string = "Hello World"

print(len(my_string))

Output: 11

2. Sice the string as per your choice?

my_string = "Hello World"

print(my_string[0:5])

Output:”Hello” print(my_string[6:])

Output: “World “

3. Concatenate two strings?

str1 = "Hello" str2 =

"World" print(str1 +

" " + str2)

Output: “Hello World “

4. Convert in to lower case in to uppercase character?

my_string = "hello world " print(my_string.swapcase())

Output: " HELLO WORLD”

5.Convert upper case into lower case charecters?

my_string = " HELLO WORLD " print(my_string.swapcase())

Output: "hello world"

6.Convert the charecters into Unicode.

my_char = 'A'

print(ord(my_char))

Output:65

7.Convert Unicode into character?

unicode_val = 65 print(chr(unicode_val))

Output:A

8. Check whether the given "substring"
exists in the string my_string = "Hello
World" substring = "World" print(substring
in my_string)
Output: True

9. Replace the character 'k' with 'h'
my_string = "Hello kWorld"
print(my_string.replace('k', 'h'))
Output: "Hello hWorld"

10. Pad the string with "x" at the end
my_string = "Hello" print(my_string + "x" *
5)
Output: "Helloxxxxx"

11. remove leading and trailing
whitespace or specified characters from the
string my_string = "   Hello World   "
print(my_string.strip())
Output: "Hello World"

12. split the given string in to group of
five characters my_string = "HelloWorld"
print([my_string[i:i+5] for i in range(0, len(my_string), 5)])
Output: ["Hello", "World"]

13. count total number of words
my_string = "Hello World this is a test"
print(len(my_string.split()))

Output: 6

14. Find the frequency of each characters

```
in the string my_string = "Hello World"
char_freq = {} for char in my_string:     if
char in char_freq:          char_freq[char] += 1
else:
        char_freq[char] = 1 print(char_freq)
```

Output: {'H': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'W': 1, 'r': 1, 'd': 1}

STDIN and File operators

15. get the file name from the user

```
file_name = input("Please enter the file
name: ") print(f"You entered: {python}")
```

Output:python

16. check the file exist or not import sys

```
import os
filename = input("Enter the file name: ")
print("You entered:", filename) if
os.path.exists(filename):
print("File exists")
else: print("File does not
exist")
```

Looping and File handling

17. read the contents from the file

```
filename = input("Enter the file name: ") try:
with open(filename, 'r') as file: contents =
file.read() print(contents) except
FileNotFoundError:  print("File not found")
```

```
Output: Enter the file name: python
File not found
```

18. reverse the contents from the file

```
filename = input("Enter the file name: ")
try:with open(filename, 'r') as file: contents =
file.read() reversed_contents = contents[::-1]
print(reversed_contents) except
FileNotFoundError:     print("File not found")
```

```
Output: Enter the file name: hello world
File not found
```

19. Write into the file filename =

```
input("Enter the file name: ") contents =
input("Enter the contents: ")
 with open(filename, 'w') as file:     file.write(contents)
```

```
Output: Enter the file name: w
Enter the contents: contents
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 4, in <module>
```

Math operations

20. convert Frequency in to percentage

(continuation of 12th Question) my_string =

```
"Hello World" char_freq = {} for char in
my_string: if char in char_freq:char_freq[char]
+= 1     else: char_freq[char] = 1
 total_chars = len(my_string) for char, freq in char_freq.items():     percentage =
(freq / total_chars) * 100     print(f"{char}: {percentage:.2f}%") Output:  H: 9.09% e:
9.09% l: 27.27% o: 18.18%  : 9.09% W: 9.09% r: 9.09% d: 9.09%
```

21. Perform modular arithmetic operation

```
num1 = int(input("Enter the first number: "))
```

```
num2 = int(input("Enter the second number: "))
modulus = int(input("Enter the modulus: "))
result = (num1 + num2) % modulus
print(result) Output:
Enter the first number: 2
Enter the second number: 3
Enter the modulus: 2
1
```

22. Find the prime numbers check the

given number is prime or not print the prime

numbers with the given range Output:

```
   def is_prime(n):
if n <= 1:
        return False for i in range(2,
int(n ** 0.5) + 1):if n % i ==
0:return False return True
 def find_primes(start, end):
    primes = [ ]for num in
range(start, end + 1):
        if is_prime(num):
            primes.append(num)
return primes
start = int(input("Enter the start of the range: ")) end = int(input("Enter the end of the range: "))
 primes = find_primes(start, end) print("Prime numbers in the range:", primes)
Output:
 Enter the start of the range: 1
Enter the end of the range: 50
Prime numbers in the range: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

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

```
math
 def gcd(a, b):     while b:         a, b = b, a % b     return a
 num1 = int(input("Enter the first number: ")) num2 = int(input("Enter the second number: "))
 if gcd(num1, num2) == 1:
    print("Co-prime") else:
    print("Not co-prime")
Output: Enter the first number: 33
Enter the second number: 55
Not co-prime
```

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

```
math
def find_factors(n):
    factors = []     for i in range(1,
int(math.sqrt(n)) + 1):
        if n % i == 0:
            factors.append(i)
if i != n // i:
                factors.append(n // i)
return factors

num = int(input("Enter a number: "))
print(find_factors(num))
Output: Enter a number: 456
[1, 456, 2, 228, 3, 152, 4, 114, 6, 76, 8, 57, 12, 38, 19, 24]
```

25. generate 10 random numbers

```
import random
random_numbers = [random.randint(1, 100) for _ in range(10)]
print(random_numbers)
Output: [91, 13, 6, 97, 92, 76, 64, 67, 20, 11]
```

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

Let's test if 29 is probably prime using the Miller-Rabin test.

1. Express n-1 as 2^r * d:
   - n-1 = 28 = 2^2 * 7
   - So, r = 2 and d = 7.
2. Choose a random base 'a':

    - Let's pick a = 2.

3. Compute x = a^d mod n:
    - x = 2^7 mod 29 = 12
4. Check for conditions:
    - x is not congruent to 1 or -1 mod 29.
    - Calculate x^2 mod 29 = 28.
    - x^2 is congruent to -1 mod 29.

Therefore, 29 is probably prime.