Python Assignment 5

Name : Ashadullah Danish

Roll No: 24901307

Q1. Write a program that fetches data from a specific URL and print it on screen.

```
import requests

def fetch_url_data(url):
    try:
        #Send GET request to URL
        response = requests.get(url)
        #Check if request was successful
        response.raise_for_status()
        #Print the content
        print("Data from URL:")
        print(response.text)
    except requests.RequestException as e:
        print(f"Error fetching data: {e}")

#Test the function
url = "https://api.github.com"
fetch_url_data(url)
```

```
thub.com/gists{/gist_id}","hub_url":"https://api.github.com/hub","issue_search_url":"https://api.github.com/search/issues?q={query}{&pag
```

Q2. Write a Program that computes total size of all the files in your current directory folder.

```
import os

def get_directory_size():
    total_size = 0
    #Get current directory
    current_dir = os.getcwd()

    #Iterate through all files in directory
    for filename in os.listdir(current_dir):
        filepath = os.path.join(current_dir, filename)
        if os.path.isfile(filepath):
            total_size += os.path.getsize(filepath)

    #Convert to MB for readability
    size_in_mb = total_size / (1024 * 1024)
    print(f"Total size of files: {size_in_mb:.2f} MB")

get_directory_size()
```

```
Total size of files: 0.00 MB
```

Q3. Write a program that reads a file line by line and each line read from the file is copied and to another file with line numbers specified at the beginning of the line.

```
def copy_with_line_numbers(input_file, output_file):
    try:
        with open(input_file, 'r') as source, open(output_file, 'w') as target:
            for line_num, line in enumerate(source, 1):
                #Write line number and content to new file
                target.write(f"{line_num}: {line}")
        print("File copied successfully with line numbers!")
    except FileNotFoundError:
        print("Input file not found!")
    except Exception as e:
        print(f"An error occurred: {e}")

#Test the function
input_file = "/content/Sample1.txt"
output_file = "/content/sample.txt"
copy_with_line_numbers(input_file, output_file)
```

```
File copied successfully with line numbers!
```

Q4. Write a program that counts the number of tabs, spaces and newline characters in a file.

```
def count_characters(filename):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            spaces = content.count(' ')
            tabs = content.count('\t')
            newlines = content.count('\n')

            print(f"Spaces: {spaces}")
            print(f"Tabs: {tabs}")
            print(f"Newlines: {newlines}")
    except FileNotFoundError:
        print("File not found!")

#Test the function
filename = "/content/Sample1.txt"
count_characters(filename)
```

```
Spaces: 240
Tabs: 0
Newlines: 38
```

Q5. Write a program that reads data from the file and calculates the percentage of vowels and consonants in the file.

```
def calculate_letter_percentages(filename):
    try:
        with open(filename, 'r') as file:
            content = file.read().lower()
```

```
            #Count vowels and consonants
            vowels = sum(1 for char in content if char in 'aeiou')
            consonants = sum(1 for char in content if char.isalpha() and char not in 'aeiou')

            total_letters = vowels + consonants
            if total_letters > 0:
                vowel_percentage = (vowels / total_letters) * 100
                consonant_percentage = (consonants / total_letters) * 100

                print(f"Vowels: {vowel_percentage:.2f}%")
                print(f"Consonants: {consonant_percentage:.2f}%")
            else:
                print("No letters found in file")
    except FileNotFoundError:
        print("File not found!")

#Test the function
filename = "/content/Sample1.txt"
calculate_letter_percentages(filename)
```

```
Vowels: 39.58%
Consonants: 60.42%
```

Q6. What are different access modes in which you can open the file.

File Access Modes in Python:

'r' : Read mode (default) - Opens file for reading

'w' : Write mode - Opens file for writing (creates new file/overwrites existing)

'a' : Append mode - Opens file for appending

'r+' : Read and Write mode - Opens file for both reading and writing

'w+' : Write and Read mode - Opens file for both writing and reading (overwrites file)

'a+' : Append and Read mode - Opens file for both appending and reading

'b' : Binary mode (can be combined with above modes)

't' : Text mode (default)

Q7. Write a program that writes data to a file in such a way that each character after a full stop is capitalized and all the numbers are written in brackets.

```
def process_text(input_file, output_file):
    try:
        with open(input_file, 'r') as source, open(output_file, 'w') as target:
            content = source.read()

            #Split by full stop and process
            sentences = content.split('.')
            processed_content = ''

            for i, sentence in enumerate(sentences):
```

```
                if sentence:
                    #Capitalize first character of sentence
                    sentence = sentence.lstrip().capitalize()
                    #Add brackets to numbers
                    for num in "0123456789":
                        sentence = sentence.replace(num, f"[{num}]")
                    processed_content += sentence
                    if i < len(sentences) - 1:
                        processed_content += '.'

            target.write(processed_content)
        print("File processed successfully!")
    except FileNotFoundError:
        print("Input file not found!")

#Test the function
input_file = "/content/sample.txt"
output_file = "/content/sample1.txt"
process_text(input_file, output_file)
```

```
File processed successfully!
```

Start coding or generate with AI.

Q8. Write a program that reads and copies the contents in another file. While copying replace all full stops with commas.

```
def copy_replace_periods(input_file, output_file):
    try:
        with open(input_file, 'r') as source, open(output_file, 'w') as target:
            content = source.read()
            #Replace all periods with commas
            modified_content = content.replace('.', ',')
            target.write(modified_content)
        print("File copied successfully with periods replaced!")
    except FileNotFoundError:
        print("Input file not found!")
    except Exception as e:
        print(f"An error occurred: {e}")

#Test the function
input_file = "/content/sample.txt"
output_file = "/content/sample1.txt"
copy_replace_periods(input_file, output_file)
```

```
File copied successfully with periods replaced!
```