Assignment 2

Murali Krishna Ponnam

700755557

Github : https://github.com/Muralikrishna9550/Assignment_2.git

Video : https://drive.google.com/file/d/1VTVziPpjaxT3ci4V8lpVnBLRglBLRRGU/view?usp=drive_link

1) Write a program that takes two strings from the user: first_name, last_name. Pass these variables to fullname function that should return the (full name). For example: ▪ First_name = “your first name”, last_name = “your last name” ▪ Full_name = “your full name” o Write function named “string_alternative” that returns every other char in the full_name string. Str = “Good evening” Output: Go vnn Note: You need to create a function named “string_alternative” for this program and call it from main function.

```
def completename(first_name, last_name):
    return first_name + " " + last_name
def string_alternative(full_name):
    return full_name[::2]
def main():
    first_name = input("Enter your first name: ")
    last_name = input("Enter your last name: ")
    full_name = completename(first_name, last_name)
    result = string_alternative(full_name)
    print(f"Full Name: {full_name}")
    print(f"Every Other Character in Full Name: {result}")
main()
```

Output :

```
Enter your first name: Murali
Enter your last name: Krishna
Full Name: Murali Krishna
Every Other Character in Full Name: Mrl rsn
```

2) Write a python program to find the wordcount in a file (input.txt) for each line and then print the output. o Finally store the output in output.txt file. Example: Input: a file includes two lines: Python Course Deep Learning Course Output: Python Course Deep Learning Course Word_Count: Python: 1 Course: 2 Deep: 1 Learning: 1

```python
from collections import Counter
import re

def count_words_in_line(line):
    words = re.findall(r'\b\w+\b', line)
    return words

def main():
    input_file_path = "/content/input.txt"
    output_file_path = "/content/output.txt"

    try:
        with open(input_file_path, 'r') as input_file:
            with open(output_file_path, 'w') as output_file:
                original_lines = []
                word_count = Counter()

                for line_number, line in enumerate(input_file, start=1):
                    original_lines.append(line.strip())

                    words = count_words_in_line(line)
                    word_count.update(words)



                for line in original_lines:
                    print(line)

                print("\nWord_Count:")
                for word, count in word_count.items():
                    print(f"{word}: {count}")


                output_file.write('\n'.join(original_lines) + '\n\nWord_Count:\n')
                for word, count in word_count.items():
                    output_file.write(f"{word}: {count}\n")

    except FileNotFoundError:
        print(f"Error: File '{input_file_path}' not found.")

if __name__ == "__main__":
    main()
```

Output:

```
Python course
Deep learning course

Word_Count:
Python: 1
course: 2
Deep: 1
learning: 1
```

3) Write a program, which reads heights (inches.) of customers into a list and convert these heights to centimeters in a separate list using: 1) Nested Interactive loop. 2) List comprehensions Example: L1: [150,155, 145, 148] Output: [68.03, 70.3, 65.77, 67.13]

```python
# Using Nested Interactive Loop
inches_nested_loop = []
while True:
    height_inches = input("Enter height in inches (or 'done' to finish): ")
    if height_inches.lower() == "done":
        break
    inches_nested_loop.append(float(height_inches))


centimeters_nested_loop = []
for height in inches_nested_loop:

    height_cm = height * 2.54
    centimeters_nested_loop.append(round(height_cm, 2))

# Using List Comprehensions
inches_list_comp = [float(input("Enter height in inches: ")) for _ in range(len(inches_nested_loop))]


centimeters_list_comp = [round(height * 2.54, 2) for height in inches_list_comp]


print("Input Heights (in inches) - Nested Loop:", inches_nested_loop)
print("Heights Converted using Nested Loop:", centimeters_nested_loop)

print("\nInput Heights (in inches) - List Comprehension:", inches_list_comp)
print("Heights Converted using List Comprehension:", centimeters_list_comp)
```

Ouput:

```
Enter height in inches (or 'done' to finish): 150
Enter height in inches (or 'done' to finish): 155
Enter height in inches (or 'done' to finish): 145
Enter height in inches (or 'done' to finish): 148
Enter height in inches (or 'done' to finish): done
Enter height in inches: 150
Enter height in inches: 155
Enter height in inches: 145
Enter height in inches: 148
Input Heights (in inches) - Nested Loop: [150.0, 155.0, 145.0, 148.0]
Heights Converted using Nested Loop: [381.0, 393.7, 368.3, 375.92]

Input Heights (in inches) - List Comprehension: [150.0, 155.0, 145.0, 148.0]
Heights Converted using List Comprehension: [381.0, 393.7, 368.3, 375.92]
```