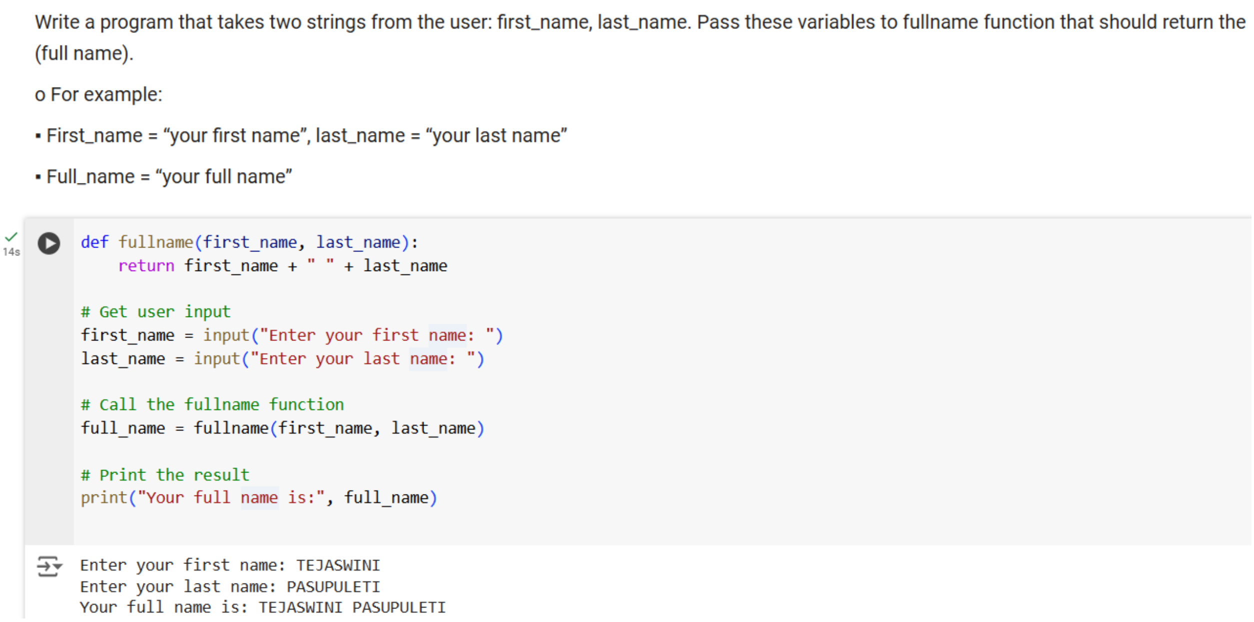
Spring 2024: CS5720

Neural Networks & Deep Learning - ICP-2

NAME: TEJASWINI PASUPULETI   ID: 700755556

GITHUB LINK: Tejaswinipasupuleti45/NN_DL_ICP2: ICP2 (github.com)

SCREENSHOTS FOR RESULTS:



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).

o For example:

▪ First_name = “your first name”, last_name = “your last name”

▪ Full_name = “your full name”

```
def fullname(first_name, last_name):
    return first_name + " " + last_name

# Get user input
first_name = input("Enter your first name: ")
last_name = input("Enter your last name: ")

# Call the fullname function
full_name = fullname(first_name, last_name)

# Print the result
print("Your full name is:", full_name)
```

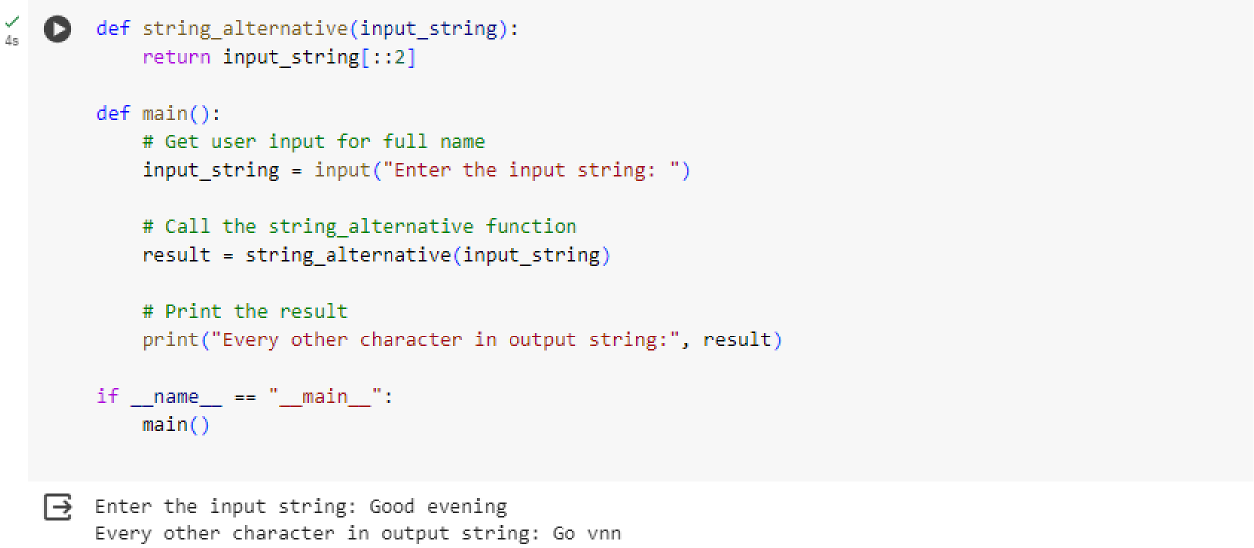
```
Enter your first name: TEJASWINI
Enter your last name: PASUPULETI
Your full name is: TEJASWINI PASUPULETI
```

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 string_alternative(input_string):
    return input_string[::2]

def main():
    # Get user input for full name
    input_string = input("Enter the input string: ")

    # Call the string_alternative function
    result = string_alternative(input_string)

    # Print the result
    print("Every other character in output string:", result)

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

```
Enter the input string: Good evening
Every other character in output string: Go vnn
```

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

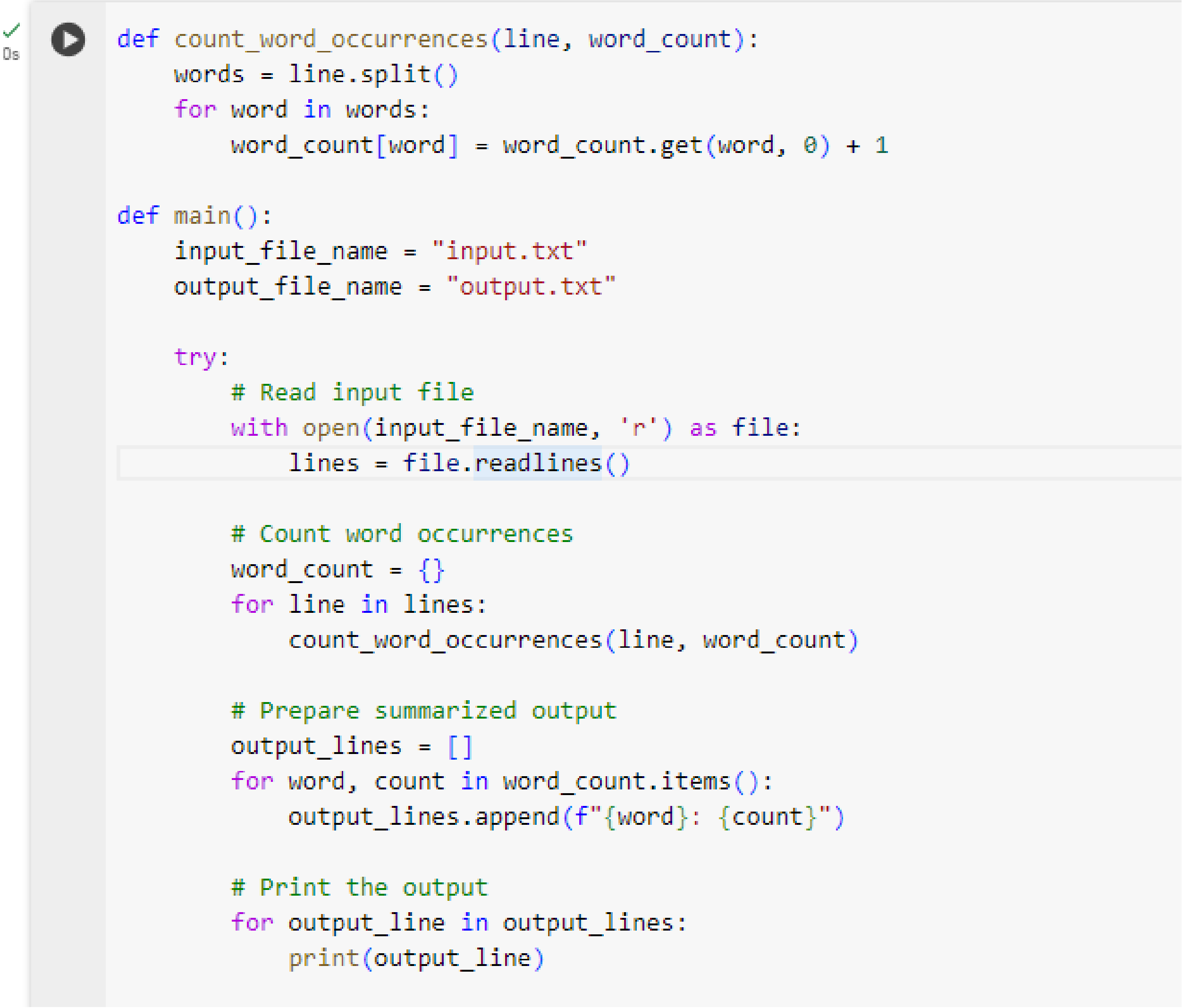
Word_Count:

Python: 1

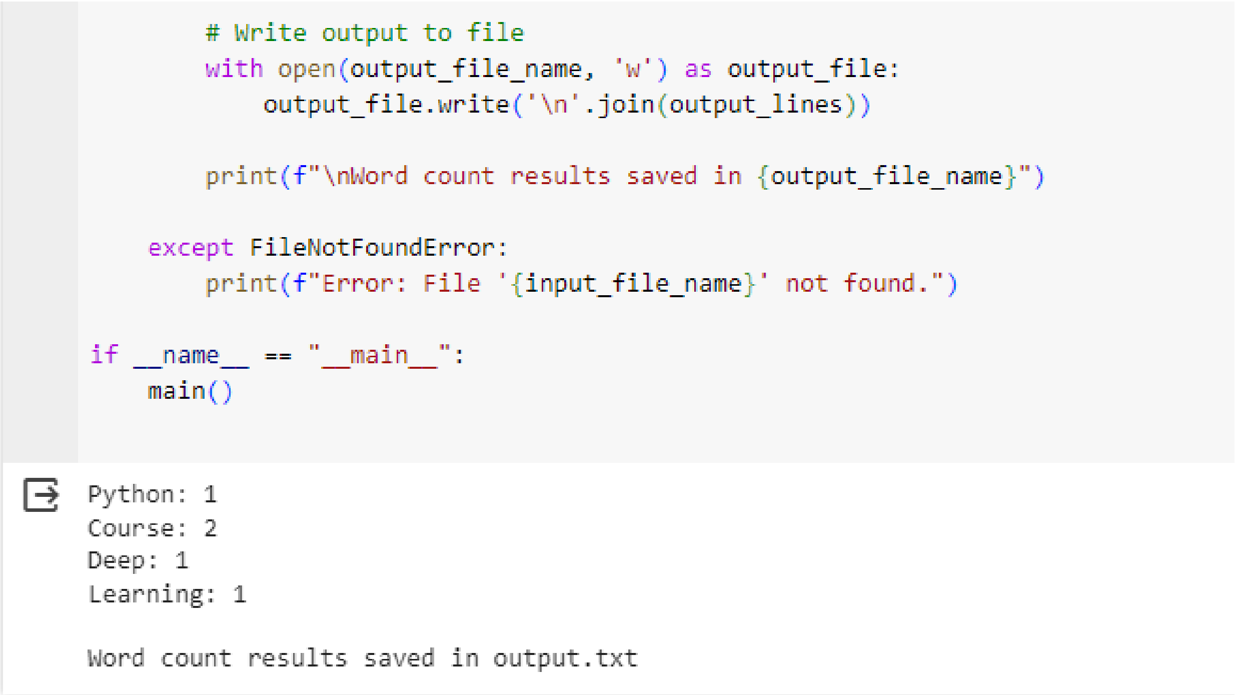
Course: 2

Deep: 1

Learning: 1

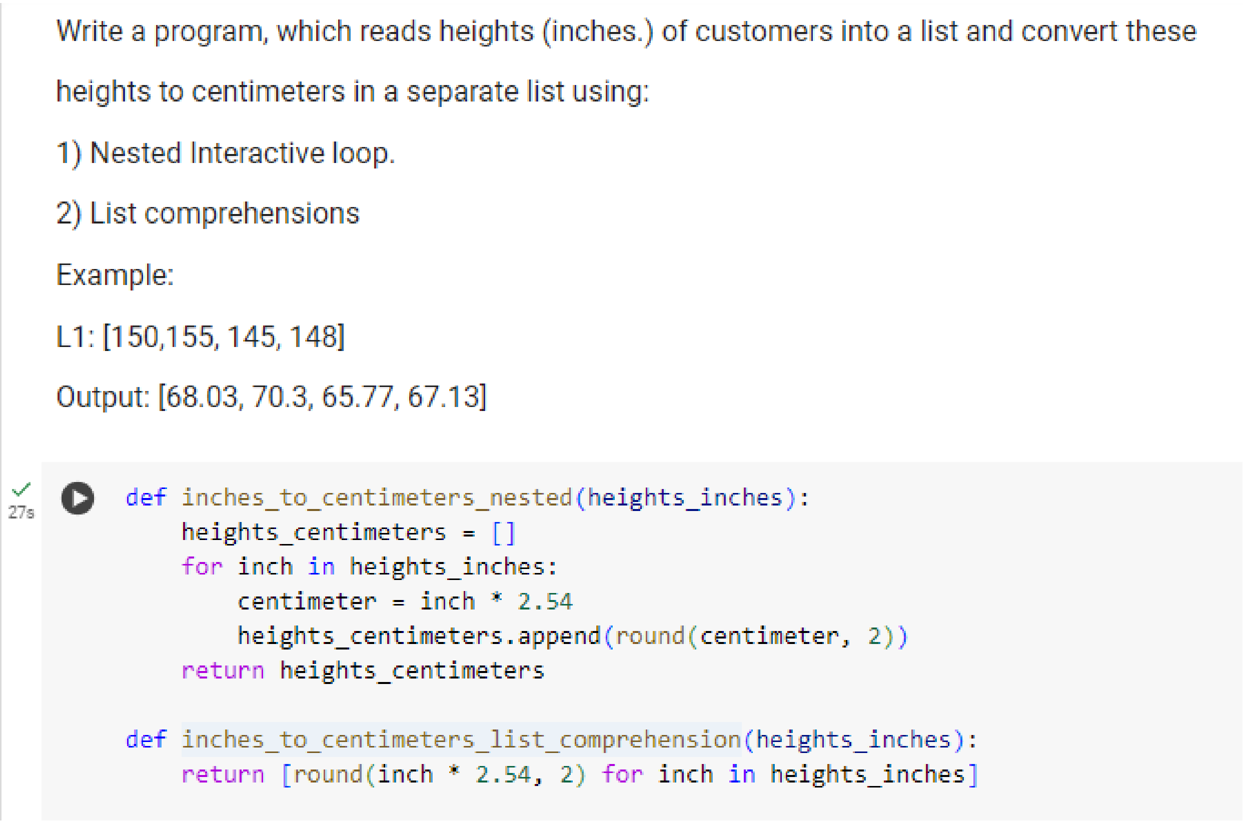


```
def count_word_occurrences(line, word_count):
    words = line.split()
    for word in words:
        word_count[word] = word_count.get(word, 0) + 1

def main():
    input_file_name = "input.txt"
    output_file_name = "output.txt"

    try:
        # Read input file
        with open(input_file_name, 'r') as file:
            lines = file.readlines()

        # Count word occurrences
        word_count = {}
        for line in lines:
            count_word_occurrences(line, word_count)

        # Prepare summarized output
        output_lines = []
        for word, count in word_count.items():
            output_lines.append(f"{word}: {count}")

        # Print the output
        for output_line in output_lines:
            print(output_line)
```



```
        # Write output to file
        with open(output_file_name, 'w') as output_file:
            output_file.write('\n'.join(output_lines))

        print(f"\nWord count results saved in {output_file_name}")

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

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

```
Python: 1
Course: 2
Deep: 1
Learning: 1

Word count results saved in output.txt
```



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

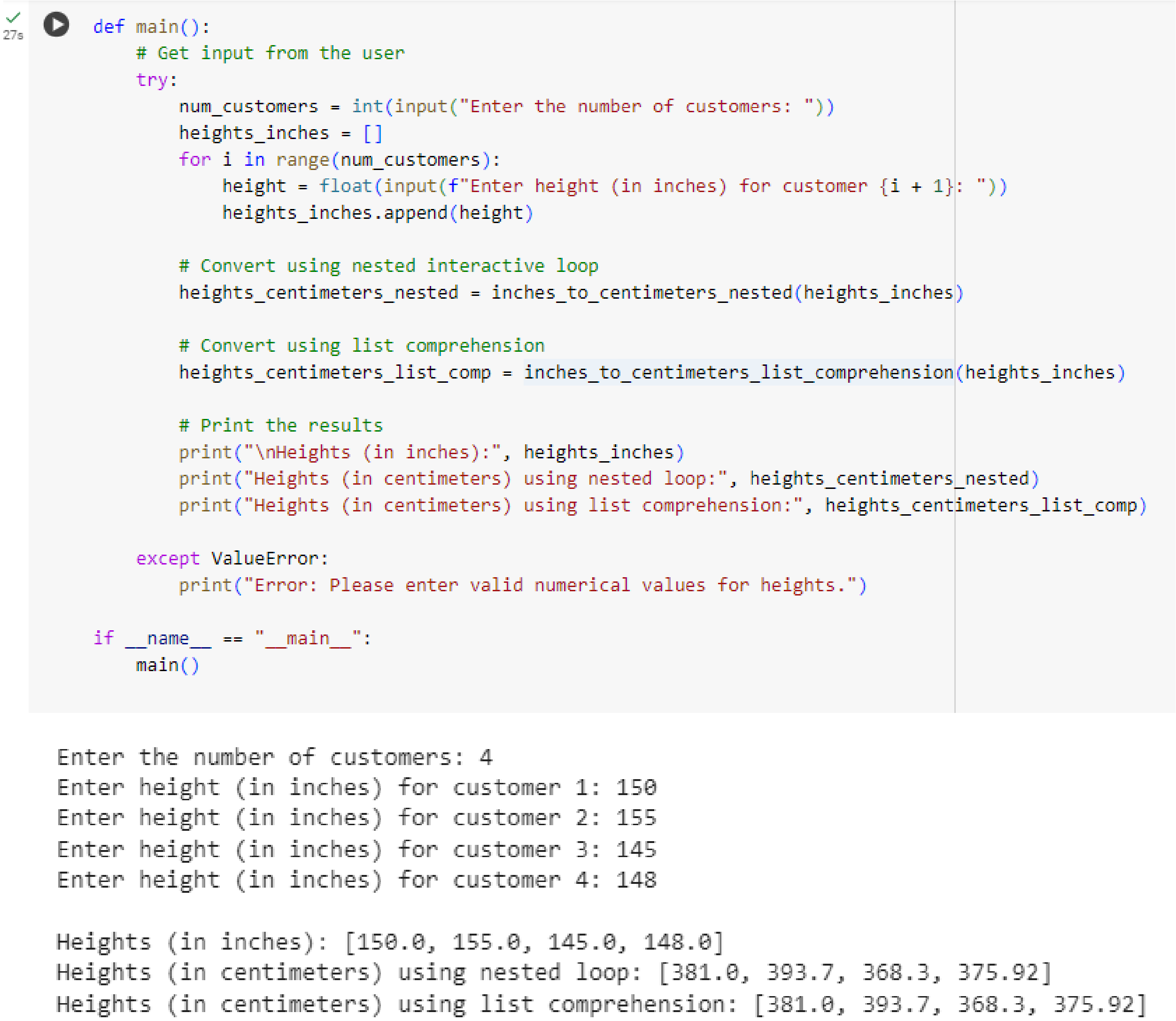
Example:

L1: [150,155, 145, 148]

Output: [68.03, 70.3, 65.77, 67.13]

✓ 27s

```
def inches_to_centimeters_nested(heights_inches):
    heights_centimeters = []
    for inch in heights_inches:
        centimeter = inch * 2.54
        heights_centimeters.append(round(centimeter, 2))
    return heights_centimeters

def inches_to_centimeters_list_comprehension(heights_inches):
    return [round(inch * 2.54, 2) for inch in heights_inches]
```



```python
def main():
    # Get input from the user
    try:
        num_customers = int(input("Enter the number of customers: "))
        heights_inches = []
        for i in range(num_customers):
            height = float(input(f"Enter height (in inches) for customer {i + 1}: "))
            heights_inches.append(height)

        # Convert using nested interactive loop
        heights_centimeters_nested = inches_to_centimeters_nested(heights_inches)

        # Convert using list comprehension
        heights_centimeters_list_comp = inches_to_centimeters_list_comprehension(heights_inches)

        # Print the results
        print("\nHeights (in inches):", heights_inches)
        print("Heights (in centimeters) using nested loop:", heights_centimeters_nested)
        print("Heights (in centimeters) using list comprehension:", heights_centimeters_list_comp)

    except ValueError:
        print("Error: Please enter valid numerical values for heights.")

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

```
Enter the number of customers: 4
Enter height (in inches) for customer 1: 150
Enter height (in inches) for customer 2: 155
Enter height (in inches) for customer 3: 145
Enter height (in inches) for customer 4: 148

Heights (in inches): [150.0, 155.0, 145.0, 148.0]
Heights (in centimeters) using nested loop: [381.0, 393.7, 368.3, 375.92]
Heights (in centimeters) using list comprehension: [381.0, 393.7, 368.3, 375.92]
```