Neural Networks & Deep Learning ICP-1

S. Ramya sri

7007727100

GitHub link: https://github.com/Ramyasri0123/ICP-1

Video Link: https://drive.google.com/file/d/1e5AXJaJedDIKa7c3kr77UipMPbjmDeQo/view?usp=drive\_link

Question 1:

**A screenshot of a computer program

Description automatically generated**

1. Complete name In order to create a full name, the function concatenates the first and last names with a space between them.

-Parameters: First\_ name (string), last\_name (string).

The output is full\_name (string), which combines first\_name and last\_name.

1. string\_alternative Function: -

Goal: Generates a new string from a given string by selecting the first character and then adding all the subsequent characters.   
The parameter full\_name (string) is used.   
The string of alternate characters from the full\_name is returned as alternate\_chars (string).

3. Main Function:

- Serves as the script's entry point when it is run.   
 -The user is prompted to input their first and last names.   
 -To combine these names, use the fullname function.   
 -The user's entire name is printed.   
 -To create a string of different characters from the whole name, call the string\_alternative function.   
 -This alternative character string is printed.

4. Script Execution Flow:   
- The script will run the `main` function if it is the main program and has not been imported as a module.   
-User input, processing (using the `fullname` and `string\_alternative` functions), and output display are all coordinated by the `main` function.

**Output:**

A black screen with white text

Description automatically generated

Question 2:

A screen shot of a computer screen

Description automatically generated

Function count\_words: This function accepts as inputs a line of text and a dictionary (`word\_counts`).   
-To extract words from the line, it employs a regular expression (`re.findall(r'\b\w+\b', line.lower())`). To make sure that only whole words are chosen, the `\b\w+\b` pattern matches word boundaries, and `line.lower()` guarantees that the procedure is case-insensitive.   
-Every word discovered is either added to the `word\_counts` dictionary or, if it already exists, its count is increased.

Reading the Input File: - The script opens and reads a file ({input.txt`) line by line

Counting Words in the Whole File: To keep the frequency of each word, the script initializes an empty dictionary called `word\_counts`.

A call to `count\_words` is then made to update `word\_counts` as it iterates over each line in `lines`.

Printing Out the Data: -Input is printed by the script after each line from the input file.   
  
Printing the Word Count: The beginning of the word count output is indicated by the printing of `Word\_Count`  
-It creates an empty set called `displayed\_words` to keep track of the words that have been printed.   
-It repeats the word extraction process for every line and determines whether or not every word has already been shown. If not, it adds the word to `displayed\_words` and prints the word along with its count.

Creating an Output File: A file called `output.txt` is created or opened by the script.   
-Input: is where the original text lines are written, followed by "Word\_Count:" with the word counts.

**Output:**

**A screenshot of a computer

Description automatically generated**

**Question 3:**

**A screen shot of a computer program

Description automatically generated**

**Explanation:**

1. Convert Centimeters to Inches:
   * The function `centimeters\_to\_inches` takes a height in centimeters and converts it to inches.
2. Read Heights from User Input:
   * The function `get\_heights` prompts the user to enter a list of heights in centimeters.
   * It uses `ast.literal\_eval` to safely convert the input string into a Python list.
   * The function checks if the input is a valid list of integers. If not, it prompts an error message and returns an empty list.
3. Convert Heights to Inches:
   * The script reads the list of heights in centimeters from the user.
   * It then converts these heights to inches using two methods:
     + A nested loop: Each height is converted individually and added to the

`heights\_in\_inches\_loop` list.

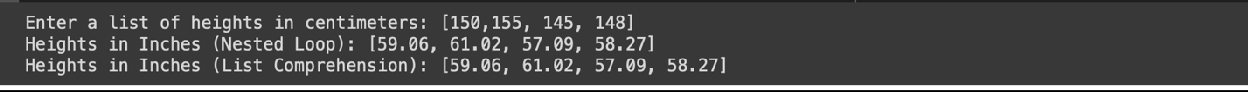
* + - List comprehension: A more concise method to achieve the same result, stored in

`heights\_in\_inches\_comprehension`.

* + Finally, the script prints the converted heights in inches using both methods for comparison.

This script effectively demonstrates two common ways to process lists in Python (a loop and list comprehension) and applies basic user input handling along with safe string evaluation using

`ast.literal\_eval`.

**Output:**