Neural Networks and Deep Learning

ICP-1

STUDENT NAME: KEERTHI YADLAPATI

STUDENT ID: 700758714

I made a video with a brief explanation. The link is given below.

https://github.com/Keerthi87140/WEEK1/blob/main/video1990044760.mp4

The Github repository link:

https://github.com/Keerthi87140/WEEK1

In the 1st question

We are asked to write a program that takes two strings from the user, I.e., first name and last name. We are asked to pass these variables to the full name function, which should return the full name. We have to write a function named string_alternative that returns every other char in the full name string.

Code:

```
def fullname(first_name, last_name):
    return f"{first_name} {last_name}"


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


def main():
    first_name = input("Enter your first name: ")
    last_name = input("Enter your last name: ")


    full_name = fullname(first_name, last_name)
    result = string_alternative(full_name)


    print(f"Full name: {full_name}")
    print(f"String with alternative characters: {result}")


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

This script focuses on two main functionalities: creating a full name from first and last names and generating a string with every alternate character.

Functions:

1. **fullname(first_name, last_name)**:
   - Takes two parameters: first_name and last_name.
   - Returns a formatted string that combines these with a space,
2. **string_alternative(input_string)**:
   - Takes one parameter: input_string.
   - Uses slicing (input_string[::2]) to extract every second character from the string.

Main Flow:

1. Prompts the user to input their first and last name.
2. Constructs the full name using the fullname function.
3. Applies the string_alternative function to the full name.
4. Prints:
   - The full name.
   - The modified string with alternative characters.

INPUT:

```
Enter your First Name: KEERTHI
Enter your Last Name: YADLAPATI
```

OUTPUT:

Full name: KEERTHI YADLAPATI

String with alternative characters: KETIYDAAI

In the 2nd question,

We had an input given with two lines

We should write a program to find the word count in a file(input.txt) for each line and print the output

CODE:

```
def count_words_in_file(input_file, output_file):
    try:
        with open(input_file, 'r') as infile:
            lines = infile.readlines()

        word_count = {}

        for line in lines:
            print(line.strip())
            words = line.strip().split()
            for word in words:
                word_count[word] = word_count.get(word, 0) + 1

        with open(output_file, 'w') as outfile:
            outfile.writelines(lines)
            outfile.write("\nWord_Count:\n")
            for word, count in word_count.items():
                outfile.write(f"{word}: {count}\n")
        print(f"\nWord counts saved to '{output_file}'.")


    except FileNotFoundError:
        print(f"Error: The file '{input_file}' does not exist.")
    except Exception as e:
        print(f"An error occurred: {e}")


input_file = "input.txt"
output_file = "output.txt"


count_words_in_file(input_file, output_file)
```

This script performs word counting in a text file and saves the results and original content to a new file.

Functionality:

1. **count_words_in_file(input_file, output_file)**:
    - Opens input_file in read mode and reads its content line by line.
    - Processes each line to split it into words, then updates a dictionary (word_count) to keep track of the occurrences of each word.
    - Writes:
        - Original content to output_file.
        - Word counts in the format {word}: {count}.
    - Handles errors such as:
        - Missing input file (FileNotFoundError).
        - General exceptions, printing an error message.

INPUT:

```
Python Course
Deep Learning Course
```

OUTPUT:

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

In the 3rd question,

We have to write a program which reads heights of customers in inches and convert these heights to centimeters.

CODE:

```
def inches_to_cm(heights):
    return [round(height * 2.54, 2) for height in heights]


def cm_to_inches(heights):
    return [round(height / 2.54, 2) for height in heights]


def main():
    print("Choose the conversion type:")
    print("1: Inches to Centimeters")
    print("2: Centimeters to Inches")
    choice = input("Enter 1 or 2: ").strip()

    if choice not in ['1', '2']:
        print("Invalid choice. Please restart the program.")
        return

    input_heights = input("Enter heights separated by commas: ")
    try:
        heights = [float(height.strip()) for height in input_heights.split(",")]
    except ValueError:
        print("Invalid input. Please enter numeric values separated by commas.")
        return

    if choice == '1':
        converted_heights = inches_to_cm(heights)
        print("Converted heights (in cm):", converted_heights)
    elif choice == '2':
        converted_heights = cm_to_inches(heights)
        print("Converted heights (in inches):", converted_heights)

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

This script converts heights between inches and centimeters based on user input.

Functions:

1. **inches_to_cm(heights)**:
   - Takes a list of heights in inches.
   - Converts each height to centimeters using the formula: cm=inches×2.54\text{cm} = \text{inches} \times 2.54cm=inches×2.54.
   - Rounds each result to two decimal places.
2. **cm_to_inches(heights)**:
   - Takes a list of heights in centimeters.
   - Converts each height to inches using the formula: inches=cm/2.54\text{inches} = \text{cm} / 2.54inches=cm/2.54.
   - Rounds each result to two decimal places.

Main Flow:

1. Asks the user to choose the conversion type (1 for inches to cm, 2 for cm to inches).
2. Prompts the user to input a comma-separated list of heights.

3. Validates and converts the input to a list of floats.
4. Applies the corresponding conversion function.
5. Prints the converted heights.

INPUT:

```
Choose the conversion type:
1: Inches to Centimeters
2: Centimeters to Inches
Enter 1 or 2: 2
Enter heights separated by commas: 155,150,145,148
```

OUTPUT:

```
Converted heights (in inches): [61.02, 59.06, 57.09, 58.27]
```