Neural Networks & Deep Learning ICP-1

Venkata Suraj, Gamini
700744962

Question-1a

```python
[2]  # Take input from the console
     input_string = input("Enter a string: ")

     # Check if the input string has at least 2 characters
     if len(input_string) < 2:
         print("Input string must have at least 2 characters.")
     else:
         # Delete at least 2 characters
         deleted_string = input_string[2:]

         # Reverse the resultant string
         reversed_string = deleted_string[::-1]

         # Print the reversed string
         print(reversed_string)
```

Explanation:

1. It takes user input as a string and stores it in the variable `input_string`.
2. It checks if `input_string` has at least 2 characters, displaying an error message if not.
3. If the input is valid, it removes at least 2 characters from the beginning of the string.
4. It then reverses the remaining characters.
5. Finally, it prints the reversed string.

Output:

```
Enter a string: Python
noht
```

Question-1b:

```python
# Take two numbers from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Perform arithmetic operations
addition_result = num1 + num2
subtraction_result = num1 - num2
multiplication_result = num1 * num2

# Check if the second number is not zero to avoid division by zero error
if num2 != 0:
    division_result = num1 / num2
else:
    division_result = "Division by zero is not allowed."

# Print the results
print(f"Addition: {num1} + {num2} = {addition_result}")
print(f"Subtraction: {num1} - {num2} = {subtraction_result}")
print(f"Multiplication: {num1} * {num2} = {multiplication_result}")
print(f"Division: {num1} / {num2} = {division_result}")
```

Explanation:

1. It takes two numbers as input from the user.
2. It performs addition, subtraction, multiplication, and division operations on these numbers.
3. It checks for division by zero and provides a message if the second number is zero.
4. It prints the results along with the corresponding mathematical expressions.

Output:

```
Enter the first number: 5
Enter the second number: 10
Addition: 5.0 + 10.0 = 15.0
Subtraction: 5.0 - 10.0 = -5.0
Multiplication: 5.0 * 10.0 = 50.0
Division: 5.0 / 10.0 = 0.5
```

Question -2:

```python
# Take input sentence from the user
input_sentence = input("Enter a sentence: ")

# Replace 'python' with 'pythons' in the sentence
output_sentence = input_sentence.replace('python', 'pythons')

# Print the modified sentence
print("Modified sentence:")
print(output_sentence)
```

Explanation:

1. It starts by taking a sentence as input from the user.
2. It searches the input sentence for the word 'python' and, whenever it finds 'python', it replaces it with 'pythons'.
3. After replacing all occurrences of 'python' in the sentence, it stores the modified sentence in the `output_sentence` variable.
4. Finally, it prints the modified sentence, effectively showing the original sentence with 'python' replaced by 'pythons'.

Output:

```
Enter a sentence: I love playing with python
Modified sentence:
I love playing with pythons
```

Question-3:

```python
[5] # Get the class score from the user
    class_score = float(input("Enter the class score: "))

    # Determine the letter grade based on the grading scheme
    if class_score >= 90:
        grade = "A"
    elif class_score >= 80:
        grade = "B"
    elif class_score >= 70:
        grade = "C"
    elif class_score >= 60:
        grade = "D"
    else:
        grade = "F"

    # Print the letter grade
    print(f"Your letter grade for the class score {class_score} is: {grade}")
```

Explanation:

1. User Input:
   - The code begins by prompting the user to input their class score as a floating-point number (e.g., 92.5). This score is stored in the `class_score` variable.

2. Determine Letter Grade:
   - The code uses a series of conditional statements (`if`, `elif`, `else`) to determine the letter grade based on the class score.

3. Print the Result: Finally, the code prints a message that includes both the original class score provided by the user and the determined letter grade. The message format is: "Your letter grade for the class score [class_score] is: [grade]."

Output:

```
Enter the class score: 69
Your letter grade for the class score 69.0 is: D
```