Spring 2024: CS5720

Neural Networks & Deep Learning - ICP-2

Assignment-1

NAME:NANDIMANDALAM VENKATA VINAY VARMA

STUDENT ID:700745193

Github Link: https://github.com/venkatavinayvarma/NeuralNetworks_ICP2.git

Video Link: https://drive.google.com/drive/folders/1B0X1eq38WGeVXGh2-kyPpdM1e71SFWM5?usp=sharing

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). o For example: ▪ First_name = “your first name”, last_name = “your last name” ▪ Full_name = “your full name”

```
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”
```

```
[3]: def fullname(first_name, last_name):
       full_name = f"{first_name} {last_name}"   #function takes two strings as input and returns the full name
       return full_name


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


     full_name = fullname(first_name, last_name) # Call the fullname function
     print(f"Full name: {full_name}") # print the result
```

```
Enter your first name:  vinayvarma
Enter your last name:  Nandimandalam
Full name: vinayvarma Nandimandalam
```

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.

```
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.
```

```
[4]: def string_alternative(string):
         l=''
         for i in range(0,len(string)):   #iterating the string from 0 to the lenght of the string
             if(i%2==0):# takes alternate characters in the string
                 l=l+string[i]  # adding empty string to the list of characters according the index
         return l

     # Example usage
     name = "Good evening"
     alternative_string = string_alternative(name) # Call the function
     print(f"Alternative string: {alternative_string}")  # print the result
```

```
Alternative string: Go vnn
```

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

```
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
```

[13]:
```
with open('input.txt','r') as ipf: #created a file named input_file
    line=ipf.read()         #used read function to read
    word=line.split()       # split functions to split the words into several words as per the question
    with open('output_file.txt','w') as opf:
        for i in word:      # iterated through word variable where the split of words are returned
            opf.write(i+':'+str(word.count(i))+'\n')
opf=open('output_file.txt','r') #opens the output file in read mode,
print(opf.read())           #reads entire file and prints it
```

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

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.

•[11]:
```
def convert_heights_interactive():
    heights_inches = []    # Create empty lists to store heights in inches and centimeters
    heights_cm = []

    num_customers = int(input("Enter the number of customers: "))    # Get the number of customers from the user

    for i in range(num_customers):  # outer loop to get heights in inches from the user
        height_inch = float(input(f"Enter height of customer {i+1} in inches: "))
        heights_inches.append(height_inch)
    for height_inch in heights_inches:    # nested loop to convert heights from inches to centimeters
        height_cm = height_inch * 2.54   # Conversion formula: 1 inch = 2.54 cm
        heights_cm.append(height_cm)     #append height

    print("Heights in centimeters:", heights_cm)  # Print the converted heights in centimeters

if __name__ == "__main__":  # Call the function
    convert_heights_interactive()
```

```
Enter the number of customers:  6
Enter height of customer 1 in inches:  12
Enter height of customer 2 in inches:  23
Enter height of customer 3 in inches:  34
Enter height of customer 4 in inches:  45
Enter height of customer 5 in inches:  65
Enter height of customer 6 in inches:  76
Heights in centimeters: [30.48, 58.42, 86.36, 114.3, 165.1, 193.04]
```

2) List comprehensions Example: L1: [150,155, 145, 148] Output: [68.03, 70.3, 65.77, 67.13]

```
Write a program, which reads heights (inches.) of customers into a list and convert these
heights to centimeters in a separate list using:
2) List comprehensions
```

[12]:
```
def convert_heights_list_comp():
    heights_inches = [150, 155, 145, 148]  # Example list of heights in inches
    heights_cm = [height_inch * 2.54 for height_inch in heights_inches]  # Convert heights to centimeters using list comprehension
    print("Heights in centimeters:", heights_cm) # Print the converted heights in centimeters

convert_heights_list_comp() # Call the function to demonstrate its usage
```

```
Heights in centimeters: [381.0, 393.7, 368.3, 375.92]
```