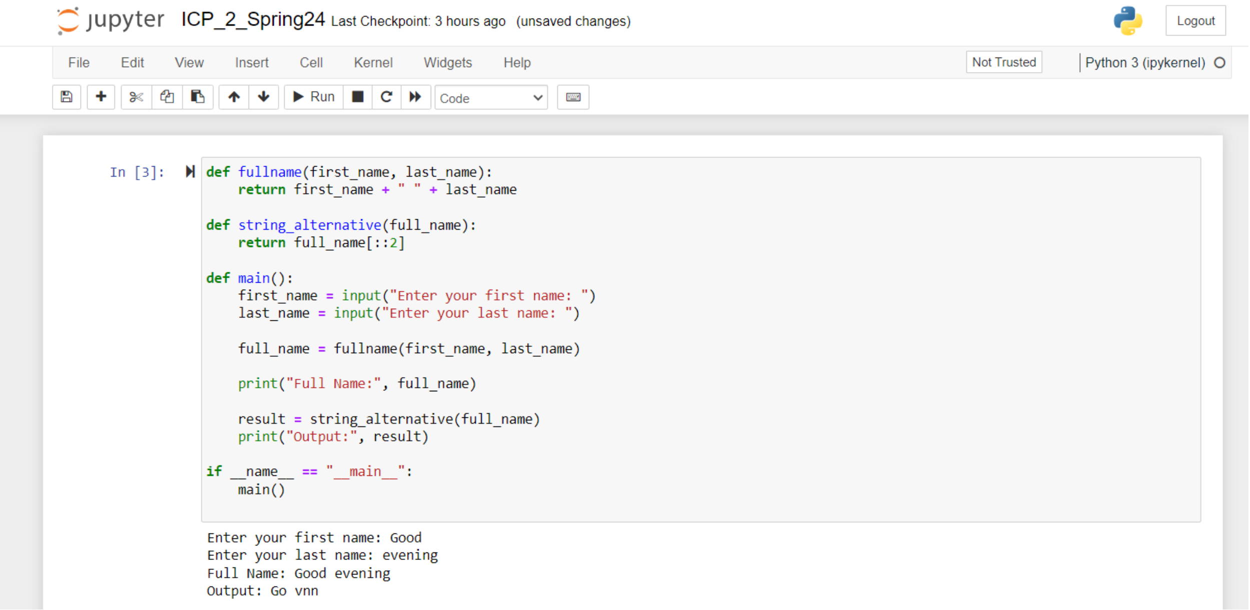
# Spring 2024: CS5720 Neural Networks & Deep Learning - ICP-2
# Bhanu Chandrika Lakkimsetti (700747439)

GitHub Link: https://github.com/bhanuchandrika99/NNDL_ICP_2

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

   Write function named “string_alternative” that returns every other char in the full_name string.
   Str = “Good evening”
   Output: Go vnn

**Solution:**



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

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

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

    full_name = fullname(first_name, last_name)

    print("Full Name:", full_name)

    result = string_alternative(full_name)
    print("Output:", result)

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

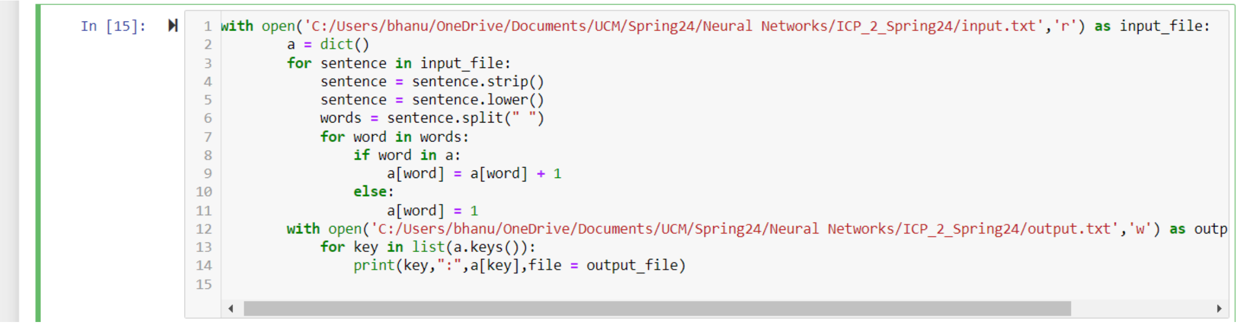
```
Enter your first name: Good
Enter your last name: evening
Full Name: Good evening
Output: Go vnn
```

2. Write a python program to find the wordcount in a file (input.txt) for each line and then print the output. Finally store the output in output.txt file.
   Example:

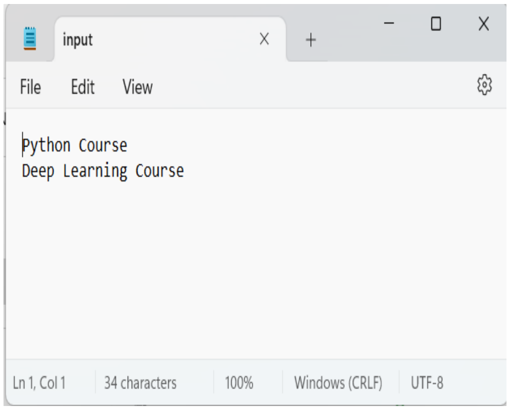
   Input: a file includes two lines:
   Python Course
   Deep Learning Course

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

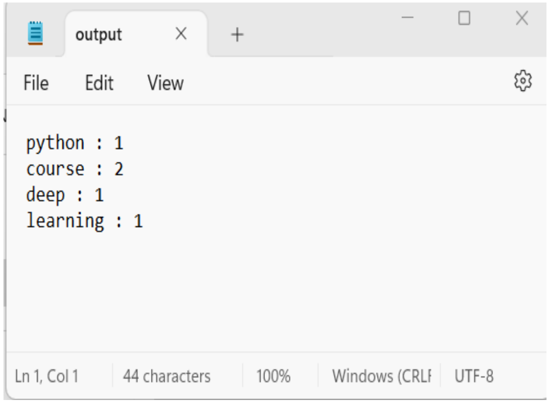
**Solution:**



```
with open('C:/Users/bhanu/OneDrive/Documents/UCM/Spring24/Neural Networks/ICP_2_Spring24/input.txt','r') as input_file:
        a = dict()
        for sentence in input_file:
            sentence = sentence.strip()
            sentence = sentence.lower()
            words = sentence.split(" ")
            for word in words:
                if word in a:
                    a[word] = a[word] + 1
                else:
                    a[word] = 1
        with open('C:/Users/bhanu/OneDrive/Documents/UCM/Spring24/Neural Networks/ICP_2_Spring24/output.txt','w') as outp
            for key in list(a.keys()):
                print(key,":",a[key],file = output_file)
```

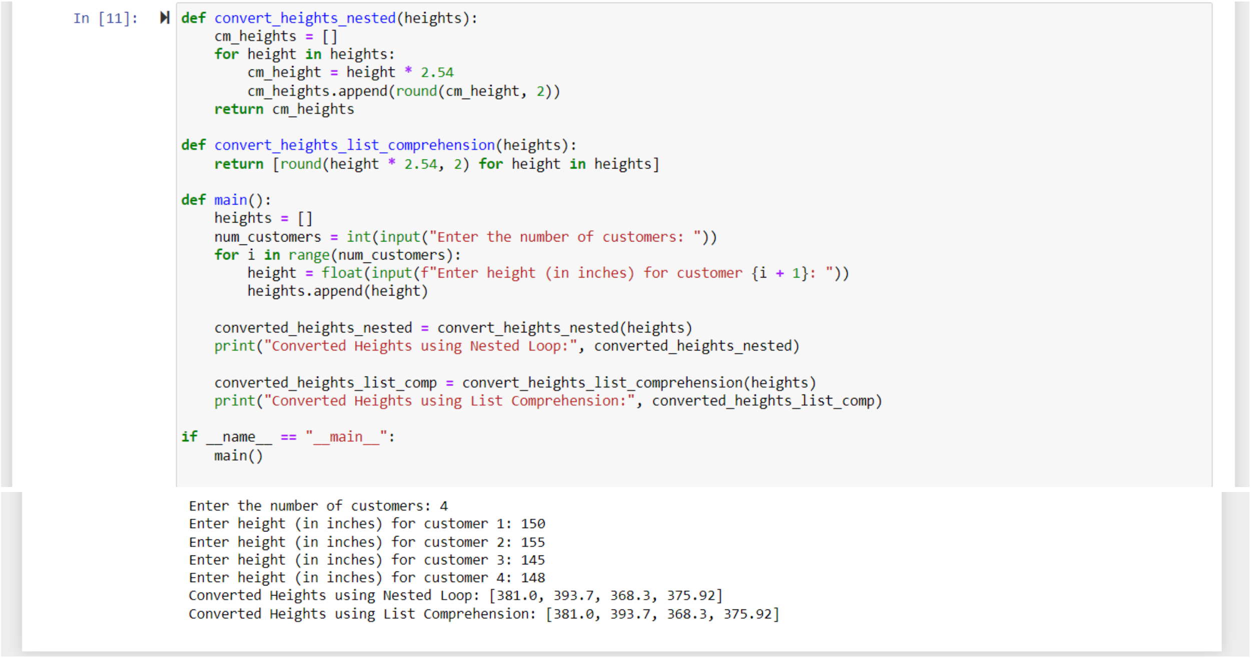


```
Python Course
Deep Learning Course
```



```
python : 1
course : 2
deep : 1
learning : 1
```

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.
2) List comprehensions



```
def convert_heights_nested(heights):
    cm_heights = []
    for height in heights:
        cm_height = height * 2.54
        cm_heights.append(round(cm_height, 2))
    return cm_heights

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

def main():
    heights = []
    num_customers = int(input("Enter the number of customers: "))
    for i in range(num_customers):
        height = float(input(f"Enter height (in inches) for customer {i + 1}: "))
        heights.append(height)

    converted_heights_nested = convert_heights_nested(heights)
    print("Converted Heights using Nested Loop:", converted_heights_nested)

    converted_heights_list_comp = convert_heights_list_comprehension(heights)
    print("Converted Heights using List Comprehension:", converted_heights_list_comp)

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
Converted Heights using Nested Loop: [381.0, 393.7, 368.3, 375.92]
Converted Heights using List Comprehension: [381.0, 393.7, 368.3, 375.92]
```