

## Neural Networks and Deep Learning

### Assignment- 02

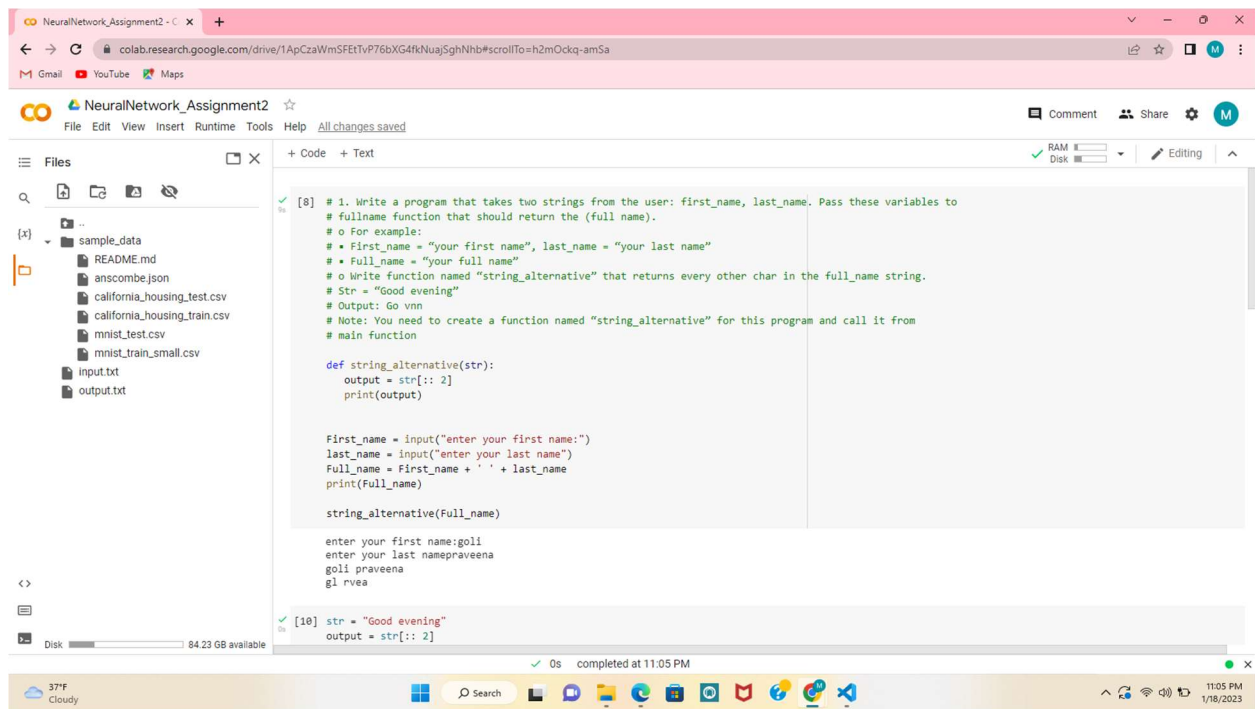
```
# 1. Write a program that takes two strings from the user: first_name, last_name.
Pass these variables to
# fullname function that should return the (full name).
# o For example:
#   First_name = "your first name", last_name = "your last name"
#   Full_name = "your full name"
# o Write function named "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

def string_alternative(str):
    output = str[:: 2]
    print(output)

First_name = input("enter your first name:")
last_name = input("enter your last name")
Full_name = First_name + ' ' + last_name
print(Full_name)

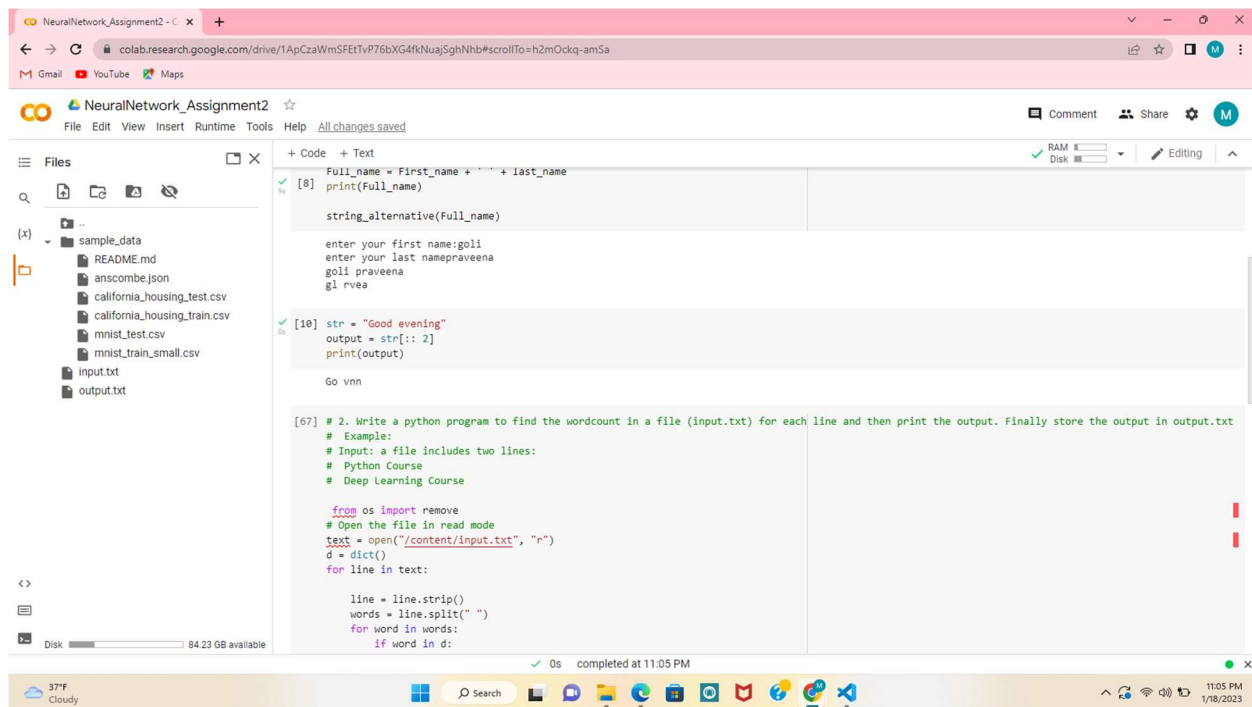
string_alternative(Full_name)
```

I have defined a function named string\_alterenative and passed str parameter and then given first name and last name from console and attached both of them as full name and then sliced the full name string so that it returns every other char in the string.



```
str = "Good evening"
output = str[:: 2]
print(output)
```

This is slicing example for the string good evening



```
# 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

from os import remove
# Open the file in read mode
text = open("/content/input.txt", "r")
d = dict()
for line in text:

    line = line.strip()
    words = line.split(" ")
    for word in words:
        if word in d:
            d[word] = d[word] + 1
        else:
            d[word] = 1

text = open("/content/input.txt", "r")
output = text.read()
f = open("/content/output.txt", "w")
```

```
f.write(output)
f.write("\n Word_Count:\n")
with open("output.txt","a") as f:
    for key, value in d.items():
        f.write('%s:%s\n' % (key, value))
```

In this I Created the input .txt file with the provided two strings in read mode and then created a empty dictionary and then removed lines and spaces between the words and created a loop to count the words repeating in the file and stored them output.txt file in write mode as keys and values where the key means words repeating in the input file and values are occurrences.

The screenshot shows a Google Colab notebook titled 'NeuralNetwork\_Assignment2'. The code in the notebook is as follows:

```
text = open("/content/input.txt", "r")
d = dict()
for line in text:
    line = line.strip()
    words = line.split(" ")
    for word in words:
        if word in d:
            d[word] = d[word] + 1
        else:
            d[word] = 1

text = open("/content/input.txt", "r")
output = text.read()
f = open("/content/output.txt", "w")
f.write(output)
f.write("\n Word_Count:\n")
with open("output.txt", "a") as f:
    for key, value in d.items():
        f.write('%s:%s\n' % (key, value))
```

The output of the program is shown in the 'output.txt' file:

```
1 Python Course
2 Deep Learning Course
3 Word_Count:
4 Python:1
5 Course:2
6 Deep:1
7 Learning:1
8
```

```
# 3. Write a program, which reads heights (inches.) customers into a list and
convert these heights to
# centimeters in a separate list using:
# 1) Nested Interactive loop.
# 2) List comprehensions
# Example: L1: [150,155, 145, 148]
# Output: [68.03, 70.3, 65.77, 67.13]

L1=[150,155, 145, 148]
cm=[]
```

```

for i in L1:
    cm.append(i*2.45)
L2 = [i*2.45 for i in L1 ]

print(cm)
print(L2)

```

here I created a list of heights in inches L1 and then I created a list of cms(centimeters) then in for loop I in the L1 I multiplied each element with 2.45 so that I can convert the inches to centimeters in cm list a list of centimeters. And created a list comprehension L2

The screenshot shows a Google Colab notebook titled "NeuralNetwork\_Assignment2". The left sidebar displays a file explorer with a folder named "sample\_data" containing several CSV and JSON files. The main code editor shows the following Python code:

```

# 3. Write a program, which reads heights (inches.) customers into a list and convert these heights to
# centimeters in a separate list using:
# 1) Nested Interactive loop.
# 2) List comprehensions
# Example: L1: [150,155, 145, 148]
# Output: [68.03, 70.3, 65.77, 67.13]

L1=[150,155, 145, 148]
cm=[]
for i in L1:
    cm.append(i*2.45)
L2 = [i*2.45 for i in L1 ]

print(cm)
print(L2)

```

The output of the code is displayed below the code cell:

```

[367.5, 379.75, 355.25, 362.6]
[367.5, 379.75, 355.25, 362.6]

```

The bottom status bar indicates the notebook is completed at 11:06 PM on 1/18/2023.

Repo : [https://github.com/Goli18/Neural\\_Assignment2.git](https://github.com/Goli18/Neural_Assignment2.git)