Spring 2025: Neural Networks & Deep Learning – ICP -1

Assignment -1

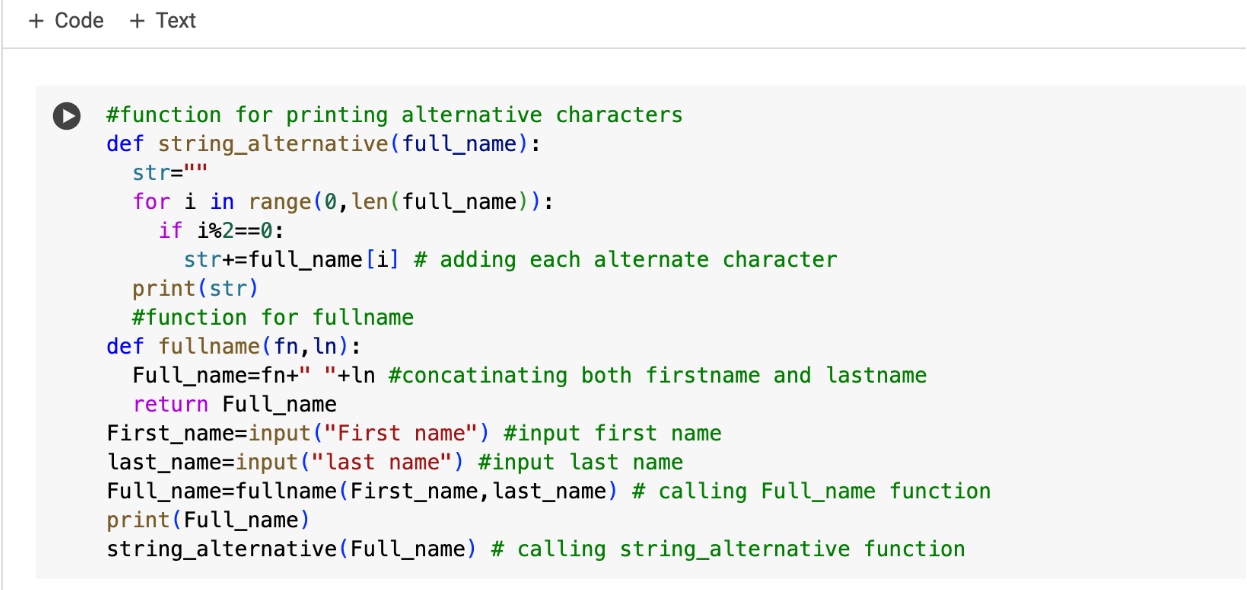
Name: Veera Manikanta Kumar Allada

Student ID: 700756934

Github Link: https://github.com/manialla da9/Neural-Networks-deep-Learning

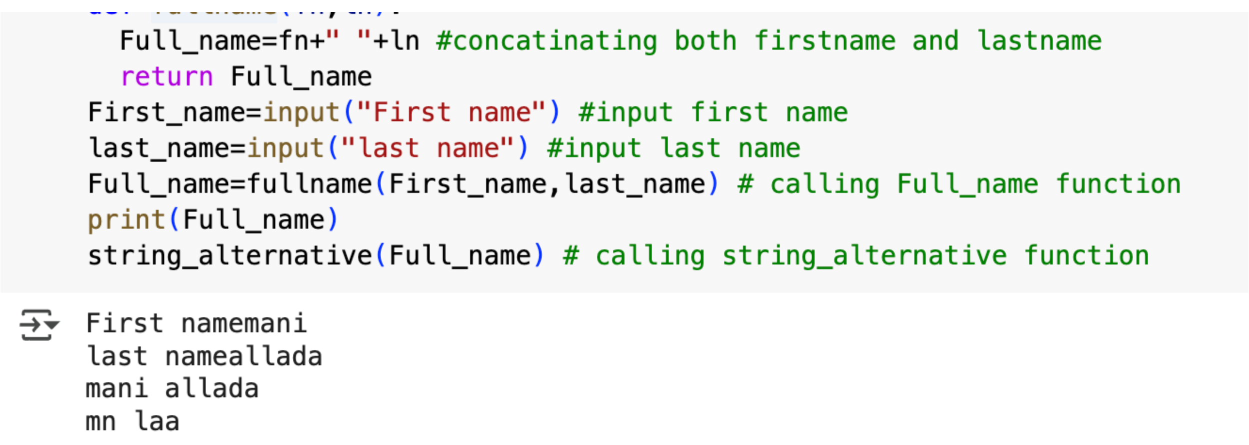
Video Link: https://drive.google.com/file/d/13ZfSG03V68urS2LqzYNM82jW8ocimckT/view?usp=drivesdk

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).



```
#function for printing alternative characters
def string_alternative(full_name):
  str=""
  for i in range(0,len(full_name)):
    if i%2==0:
      str+=full_name[i] # adding each alternate character
  print(str)
  #function for fullname
def fullname(fn,ln):
  Full_name=fn+" "+ln #concatinating both firstname and lastname
  return Full_name
First_name=input("First name") #input first name
last_name=input("last name") #input last name
Full_name=fullname(First_name,last_name) # calling Full_name function
print(Full_name)
string_alternative(Full_name) # calling string_alternative function
```

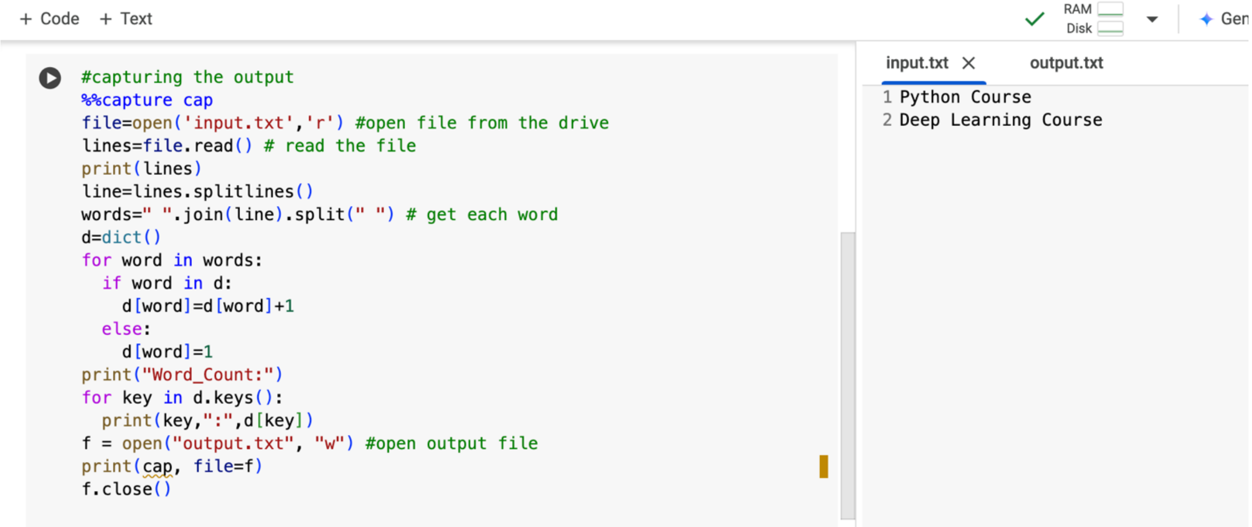
Output:



```
  Full_name=fn+" "+ln #concatinating both firstname and lastname
  return Full_name
First_name=input("First name") #input first name
last_name=input("last name") #input last name
Full_name=fullname(First_name,last_name) # calling Full_name function
print(Full_name)
string_alternative(Full_name) # calling string_alternative function
```

```
First namemani
last nameallada
mani allada
mn laa
```

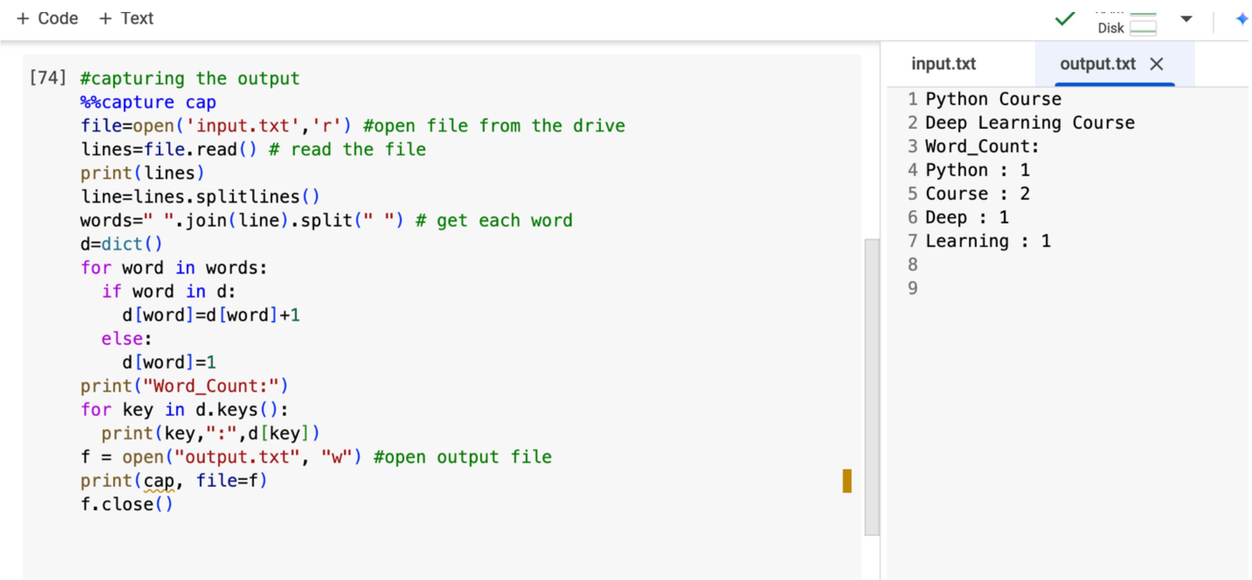
2. Write a python program to find the wordcount in a file (input.txt) for each line and then print the output.



```
#capturing the output
%%capture cap
file=open('input.txt','r') #open file from the drive
lines=file.read() # read the file
print(lines)
line=lines.splitlines()
words=" ".join(line).split(" ") # get each word
d=dict()
for word in words:
  if word in d:
    d[word]=d[word]+1
  else:
    d[word]=1
print("Word_Count:")
for key in d.keys():
  print(key,":",d[key])
f = open("output.txt", "w") #open output file
print(cap, file=f)
f.close()
```

```
Python Course
Deep Learning Course
```

Output:



```
[74] #capturing the output
%%capture cap
file=open('input.txt','r') #open file from the drive
lines=file.read() # read the file
print(lines)
line=lines.splitlines()
words=" ".join(line).split(" ") # get each word
d=dict()
for word in words:
  if word in d:
    d[word]=d[word]+1
  else:
    d[word]=1
print("Word_Count:")
for key in d.keys():
  print(key,":",d[key])
f = open("output.txt", "w") #open output file
print(cap, file=f)
f.close()
```

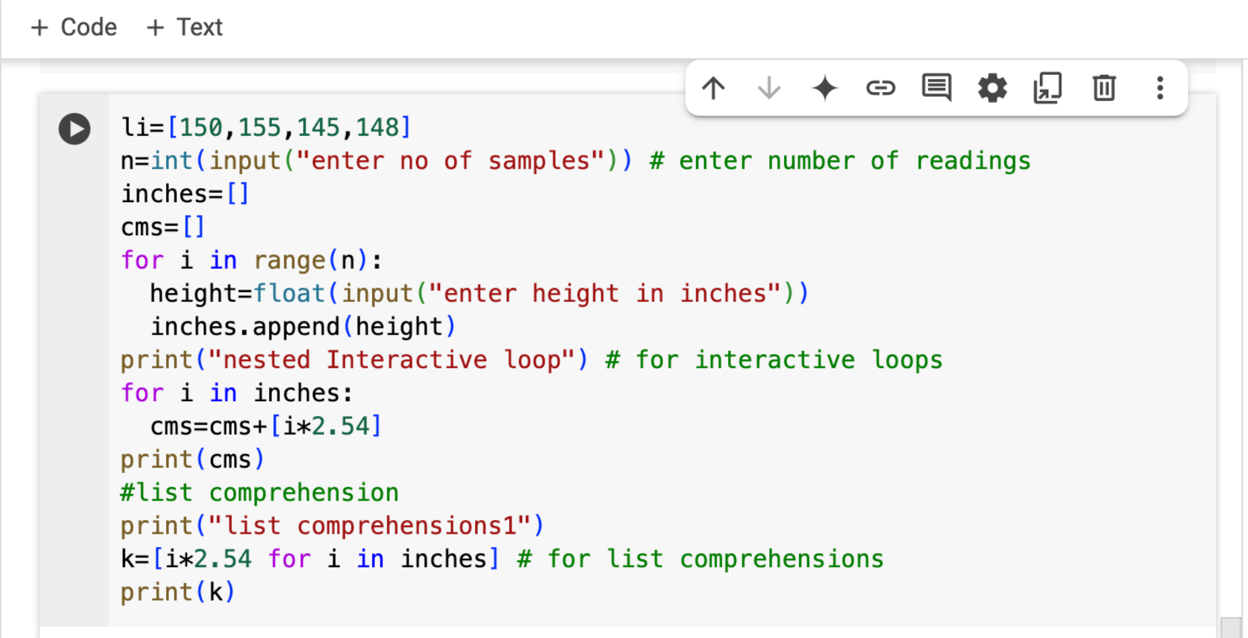
```
Python Course
Deep Learning Course
Word_Count:
Python : 1
Course : 2
Deep : 1
Learning : 1


```

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

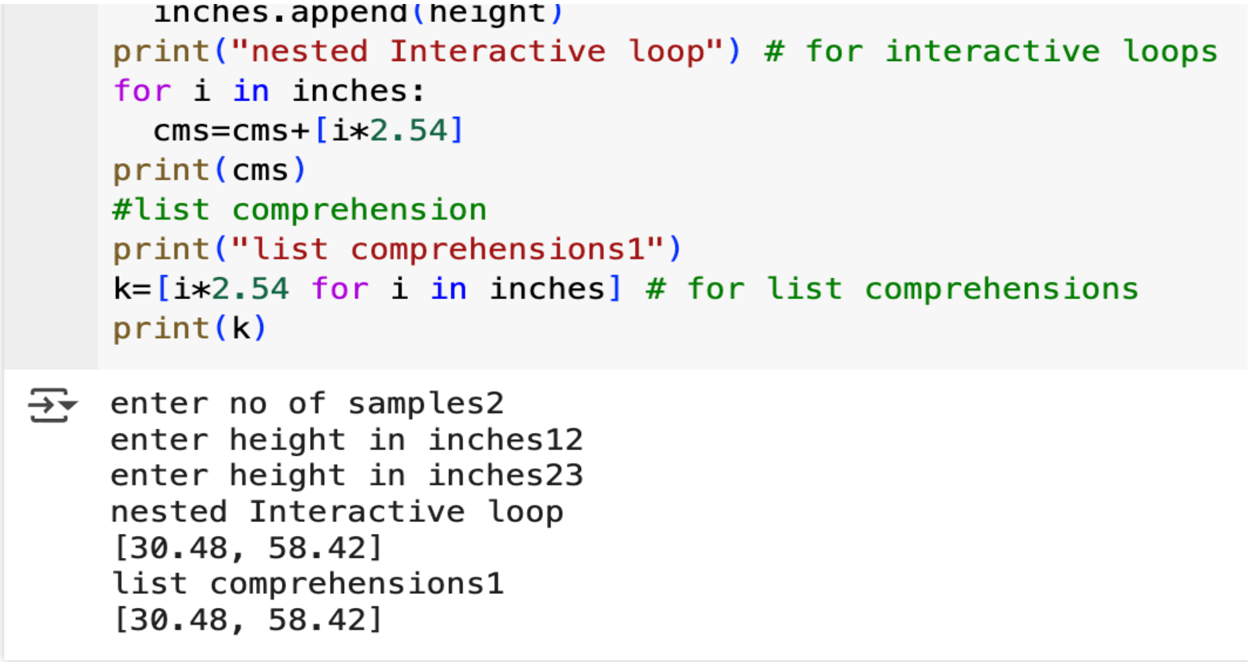
1) Nested Interactive loop.

2) List comprehensions



```
li=[150,155,145,148]
n=int(input("enter no of samples")) # enter number of readings
inches=[]
cms=[]
for i in range(n):
  height=float(input("enter height in inches"))
  inches.append(height)
print("nested Interactive loop") # for interactive loops
for i in inches:
  cms=cms+[i*2.54]
print(cms)
#list comprehension
print("list comprehensions1")
k=[i*2.54 for i in inches] # for list comprehensions
print(k)
```

Output:



```
  inches.append(height)
print("nested Interactive loop") # for interactive loops
for i in inches:
  cms=cms+[i*2.54]
print(cms)
#list comprehension
print("list comprehensions1")
k=[i*2.54 for i in inches] # for list comprehensions
print(k)
```

```
enter no of samples2
enter height in inches12
enter height in inches23
nested Interactive loop
[30.48, 58.42]
list comprehensions1
[30.48, 58.42]
```