|  |  |
| --- | --- |
| Student Name | D.Subithra |
| Student Roll Number | 962719104037 |
| Team ID | PND2022TMID52113 |

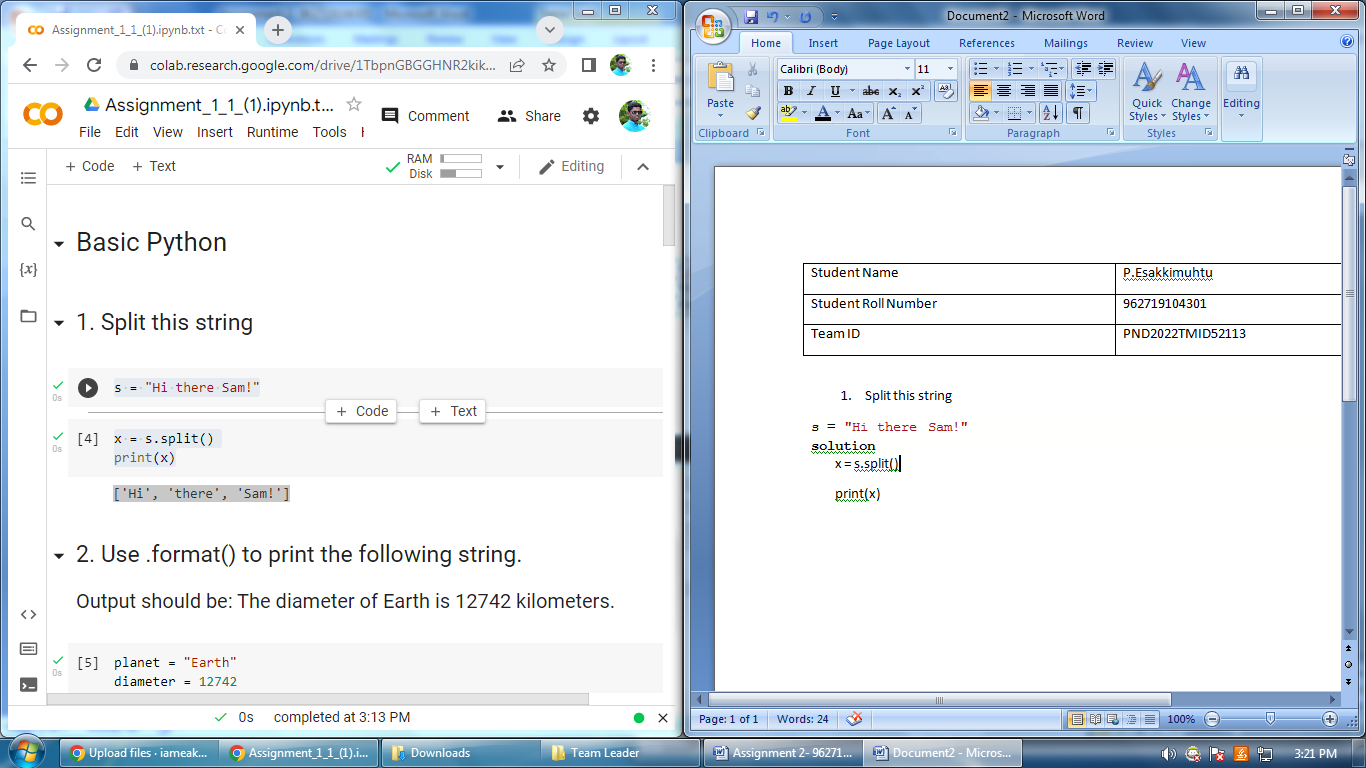
1. Split this string

s = "Hi there Sam!"

**solution**

x = s.split()

print(x)



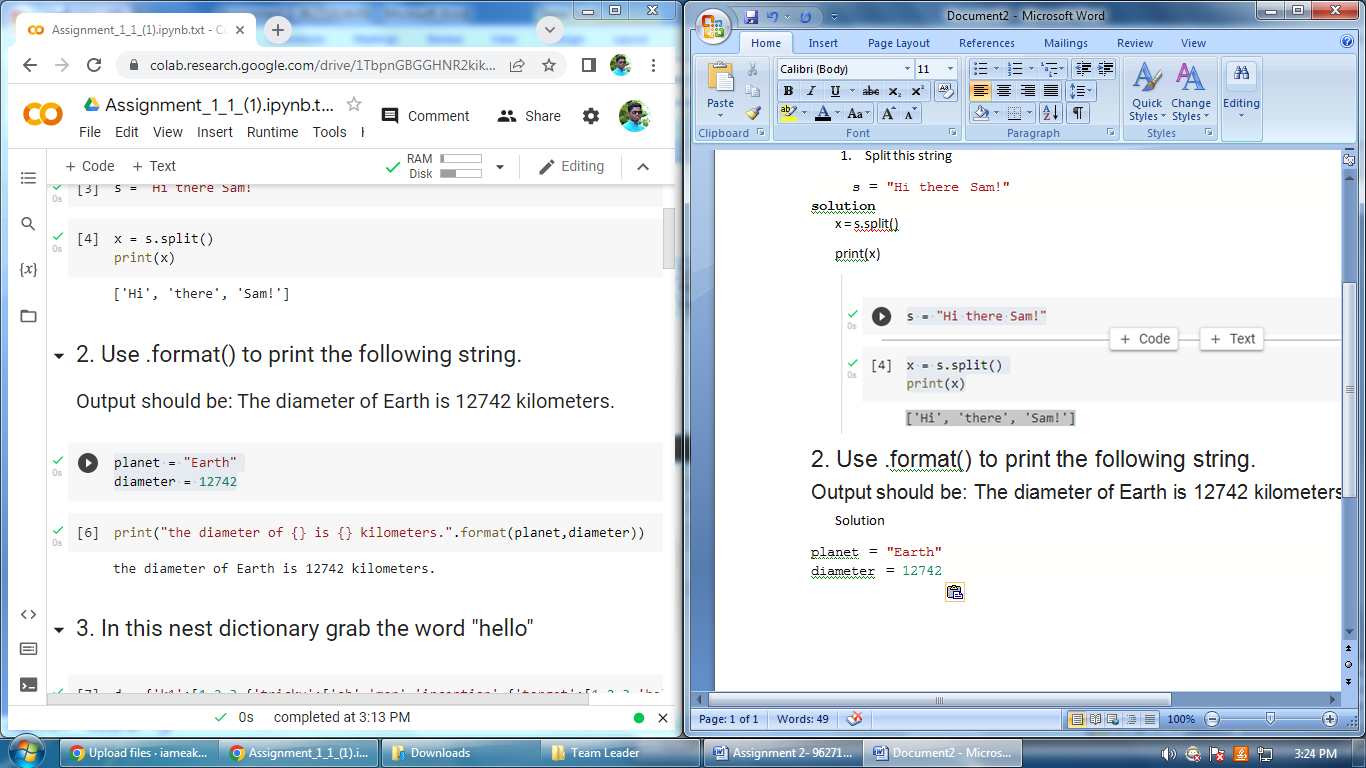
2. Use .format() to print the following string.

Output should be: The diameter of Earth is 12742 kilometers.

**Solution**

planet = "Earth"

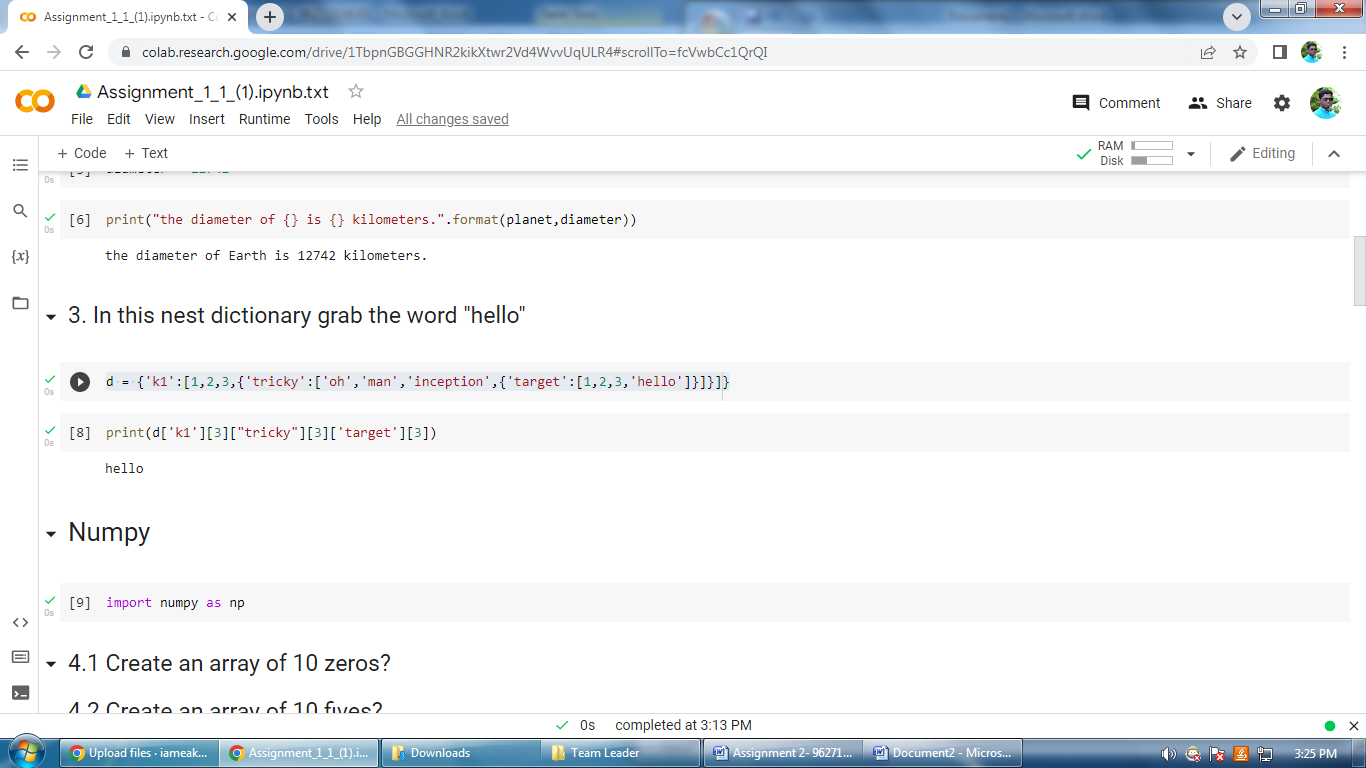
diameter = 12742



## In this nest dictionary grab the word "hello"

**solution**

d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}



## 4.1 Create an array of 10 zeros?

np.ones(10)\*0

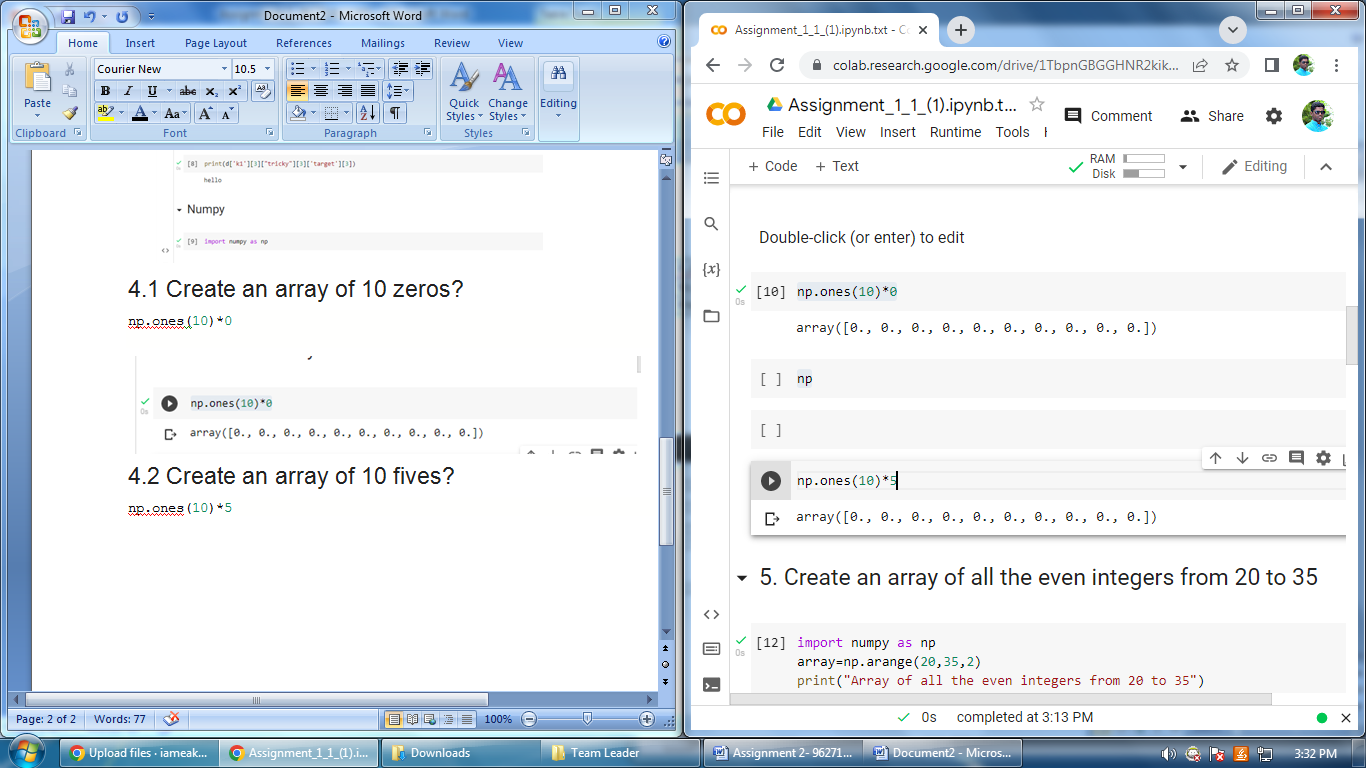
**solution**

## 

## 4.2 Create an array of 10 fives?

np.ones(10)\*5

**solution**



5. Create an array of all the even integers from 20 to 35

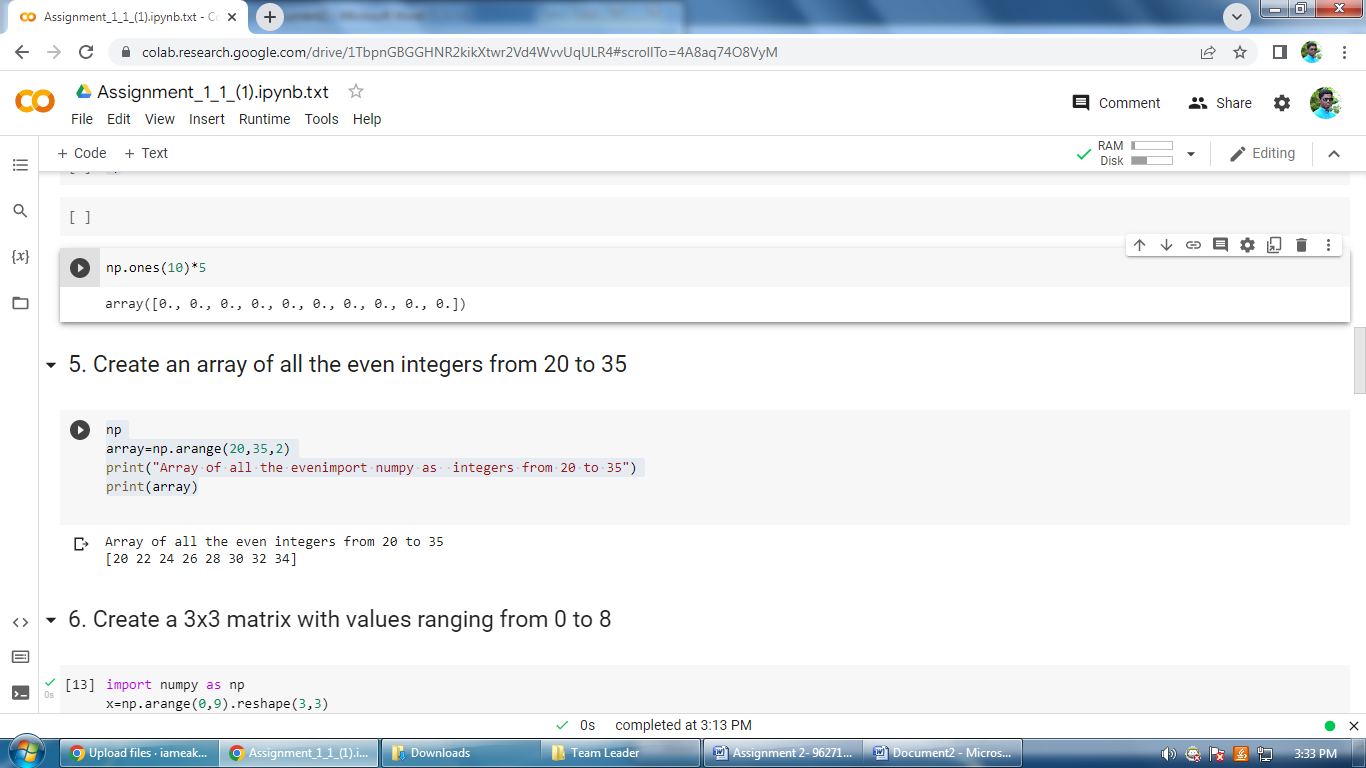
np

array=np.arange(20,35,2)

print("Array of all the evenimport numpy as  integers from 20 to 35")

print(array)

**solution**



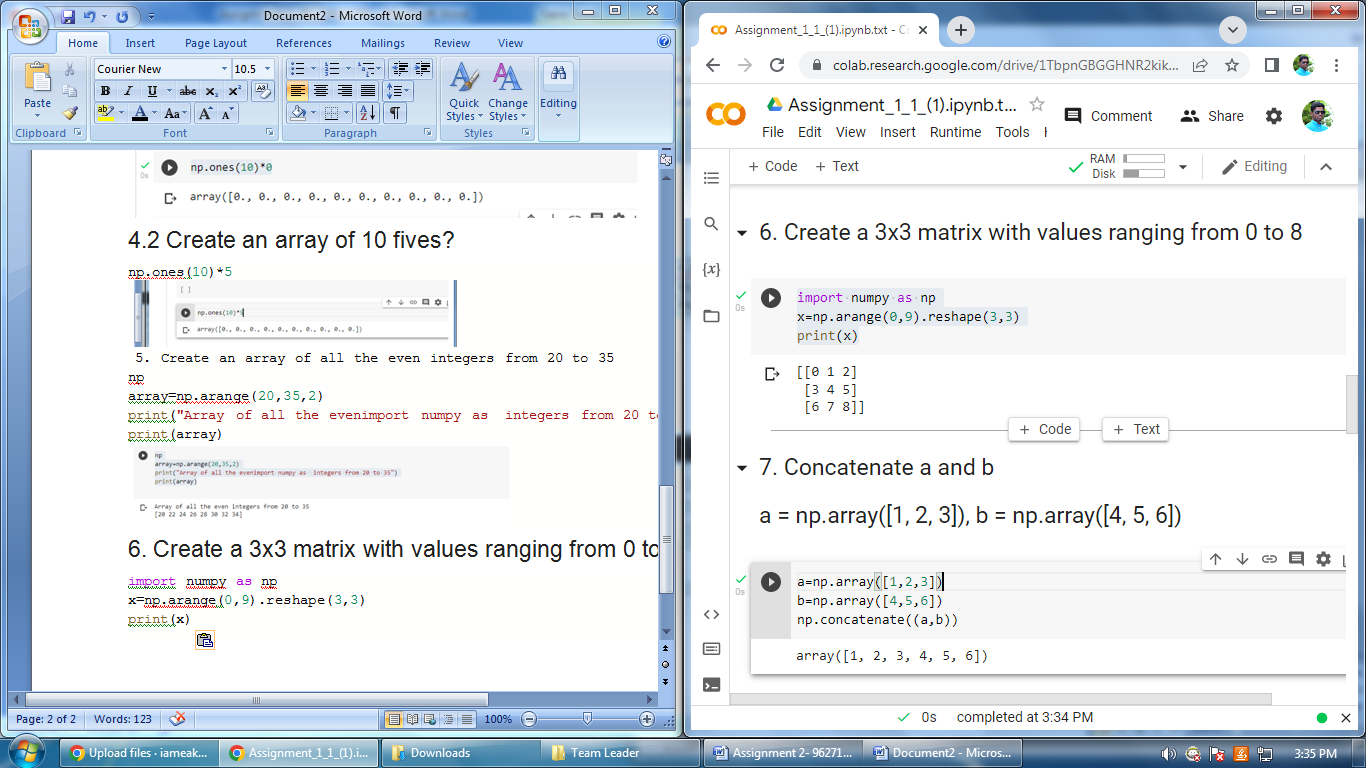
## 6. Create a 3x3 matrix with values ranging from 0 to 8

import numpy as np

x=np.arange(0,9).reshape(3,3)

print(x)

**solution**



## 7. Concatenate a and b

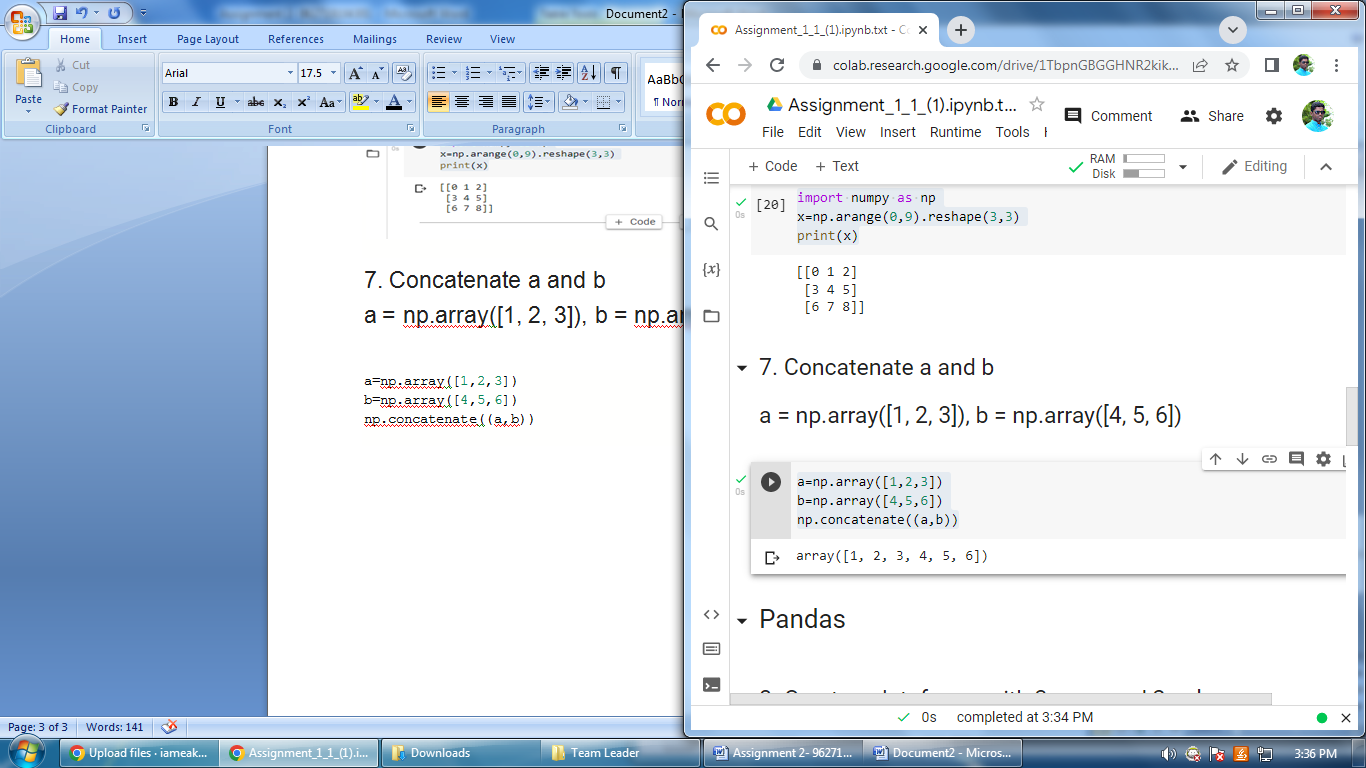
## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])

a=np.array([1,2,3])

b=np.array([4,5,6])

np.concatenate((a,b))

**solution**



8. Create a dataframe with 3 rows and 2 columns

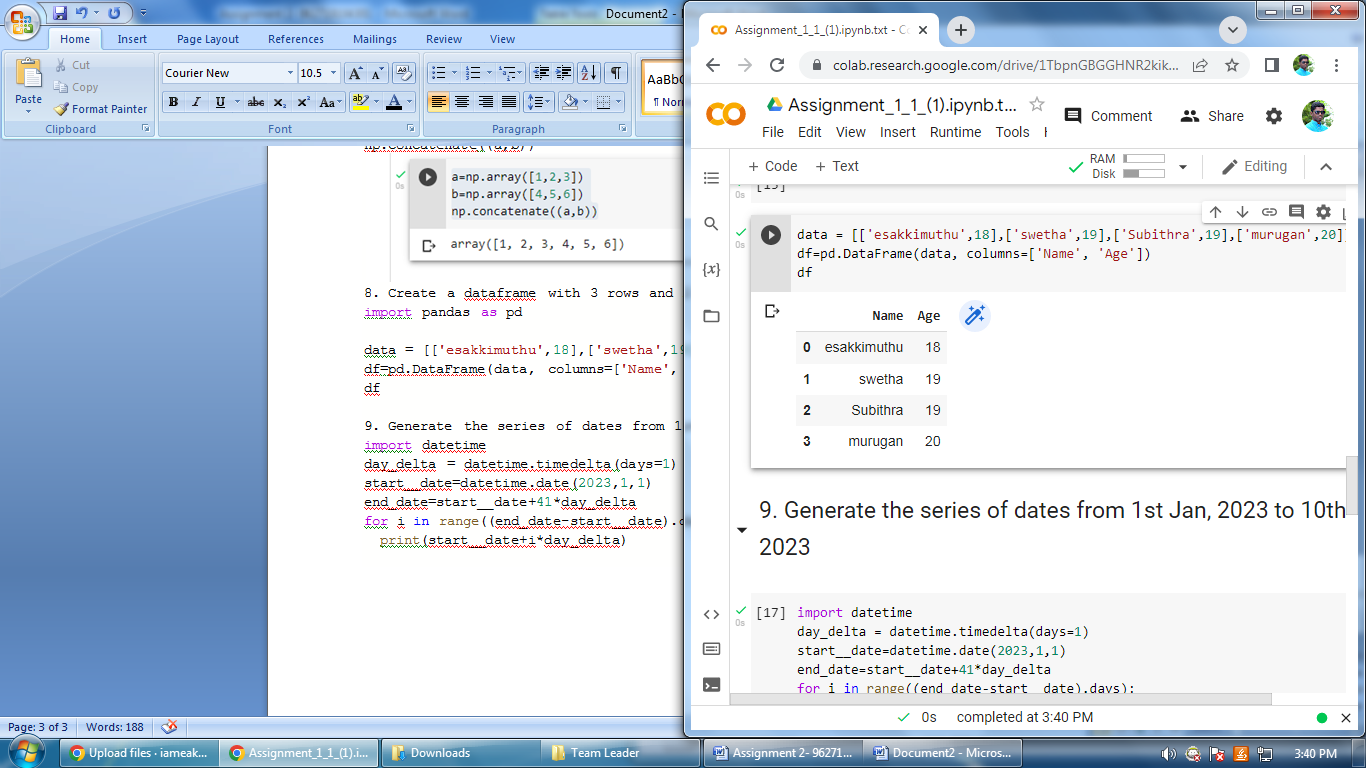
import pandas as pd

data = [['esakkimuthu',18],['swetha',19],['Subithra',19],['murugan',20]]

df=pd.DataFrame(data, columns=['Name', 'Age'])

df

**solution**



9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023

import datetime

day\_delta = datetime.timedelta(days=1)

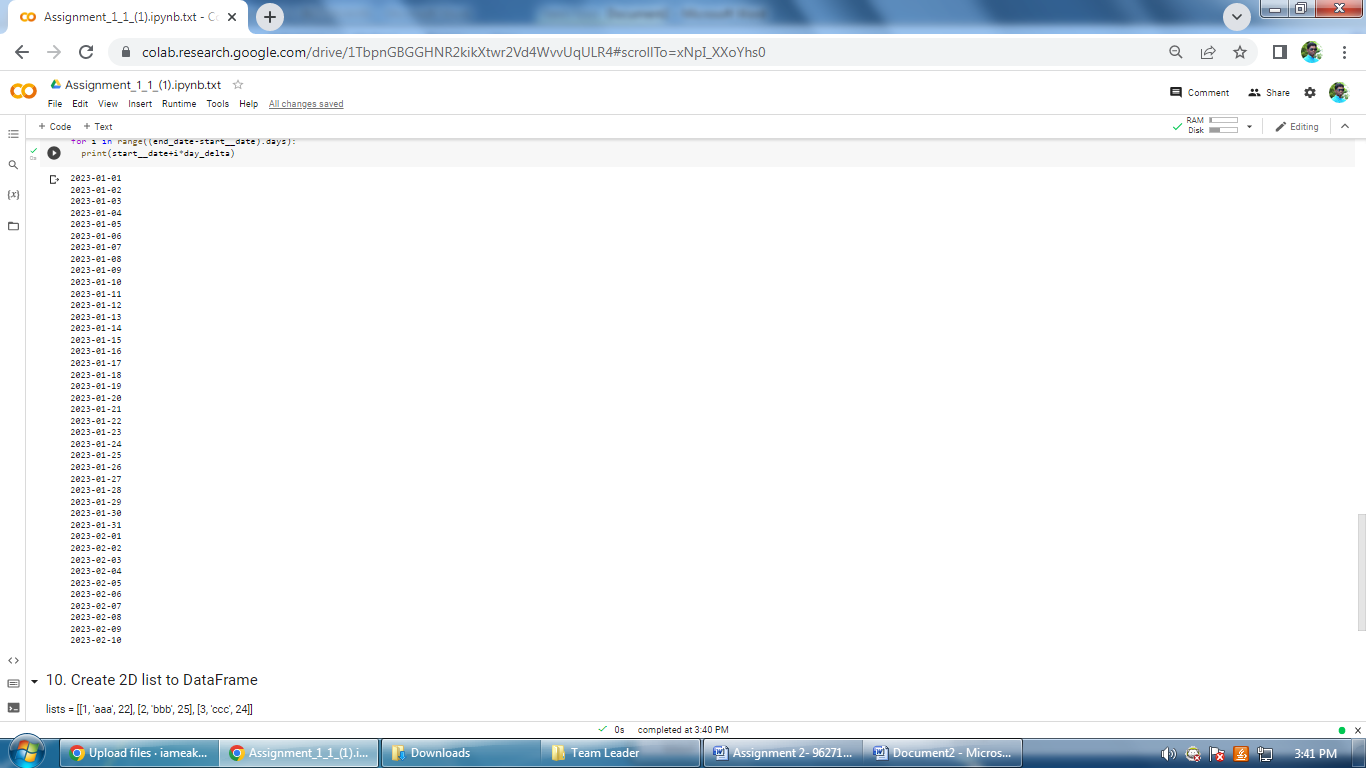
start\_\_date=datetime.date(2023,1,1)

end\_date=start\_\_date+41\*day\_delta

for i in range((end\_date-start\_\_date).days):

  print(start\_\_date+i\*day\_delta)

**solution**



## 10. Create 2D list to DataFrame

lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]

ists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]

df=pd.DataFrame(lists,columns=['Number','Letter','Number'])

print(df)

**solution**

