**DATA ANALYSIS USING PYTHON**

**LAB BOOK 1**

**HEMA V**

**18BBTCS040**

**If-Else in Python**

**Program 1**

**Objective:** To check for age above 18 using if-else statements

**CODE:**

**OUTPUT:**

age = int (input("Enter your age? "))

if age>=18:

    print("You are eligible to vote !!");

else:

    print("Sorry! you have to wait !!");

Enter your age? 25

You are eligible to vote !!

**Program 2**

**Objective:** To check for even or odd number using if-else statement

**CODE :**

**OUTPUT:**

num = int(input("Enter a number: "))

if (num % 2) == 0:

   print("{0} is Even number".format(num))

else:

  print("{0} is Odd number".format(num))

Enter a number: 5

5 is Odd number

**Program 3**

**Objective:** To check for the number which equals to 10, 50, 100 using if-else statement

**CODE & OUTPUT:**

number = int(input("Enter the number?"))

if number==10:

  print("number is equals to 10")

elif number==50:

  print("number is equal to 50");

elif number==100:

  print("number is equal to 100");

else:

  print("number is not equal to 10, 50 or 100");

Enter the number?6

number is not equal to 10, 50 or 100

**Program 4**

**Objective:** To check for the number divisible by 2

**CODE & OUTPUT:**

num = int(input("enter the number?"))

if num%2 == 0:

  print("Number is even")

print("bye")

enter the number?5

bye

**Program 5**

**Objective:** To check largest number in given 3 numbers

**CODE & OUTPUT:**

a = int(input("Enter a- "));

b = int(input("Enter b- "));

c = int(input("Enter c- "));

if a>b and a>c:

  print("a is largest");

if b>a and b>c:

  print("b is largest");

if c>a and c>b:

  print("c is largest");

Enter a- 5

Enter b- 3

Enter c- 2

a is largest

**LISTS**

**Program 1**

**Objective:** To check length of lists given

**CODE & OUTPUT:**

list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']

print ("First list length : ", len(list1))

print ("Second list length : ", len(list2))

First list length : 3

Second list length : 2

**Program 2**

**Objective:** To add the values to existing list

**CODE & OUTPUT:**

### aList = [123, 'xyz','zara','abc']; aList.append( 2009 );

print ("Updated List : ", aList)

Updated List : [123, 'xyz', 'zara', 'abc', 2009]

**Program 3**

**Objective:** To count for the occurrence of values in list

**CODE & OUTPUT:**

### aList = [123, 'xyz', 'xyz', 'abc', 123]; print ("Count for xyz : ", aList.count('xyz'))

print ("Count for zara : ", aList.count('zara'))

Count for xyz : 2 Count for zara : 0

**Program 4**

**Objective:** To add the values to an existing list using another list

**CODE & OUTPUT:**

### aList = [123, 'xyz', 'zara', 'abc', 123]; bList = [2009, 'hema']; aList.extend(bList)

print ("Extended List : ", aList)

Extended List : [123, 'xyz', 'zara', 'abc', 123, 2009, 'hema']

**Program 5**

**Objective:** To check for index values of the elements in the list

**CODE & OUTPUT:**

### aList = [123, 'xyz', 'zara', 'abc', 'zara'];

print ("Index for xyz : ", aList.index( 'xyz' ) ) print ("Index for zara : ", aList.index( 'zara'))

Index for xyz : 1 Index for zara : 2

**Program 6**

**Objective:** To insert value/element at specific index value

**CODE & OUTPUT:**

### aList = [123, 'xyz','hema','abc'] aList.insert( 3, 2009)

print ("Final List : ", aList)

Final List : [123, 'xyz', 'hema', 2009, 'abc']

**Program 7**

**Objective:** Pop function in list

**CODE & OUTPUT:**

### aList = [123, 'xyz','hem','abc']; print(aList)

print ("List after popping last element :",aList.pop()) print(aList)

print ("List after popping element from mentioned index:",aList.pop(2))

print(aList)

aList.insert(3, 2009)

print(aList)

[123, 'xyz', 'zara', 'abc']

List after popping last element :abc [123, 'xyz', 'zara']

List after popping element from mentioned index: hem [123, 'xyz']

[123, 'xyz', 2009]

**Program 8**

**Objective:** Reversing of list in python

**CODE & OUTPUT:**

### aList = ['shabnam', 'xyz', 'zara', 'abc', 'hem'];

aList.reverse();

print ("List : ", aList)

List : ['hem', 'abc', 'zara', 'xyz', 'shabnam']

**Program 9**

**Objective:** To remove the value in the list

**CODE & OUTPUT:**

### aList = [123, 'xyz', 'hem','abc','xyz']; aList.remove('xyz');

print ("List :",aList) aList.remove('abc'); print ("List :",aList)

List : [123, 'hem', 'abc', 'xyz']

List : [123, 'hem', 'xyz']

**Program 10**

**Objective:** To sort the list

**CODE & OUTPUT:**

### aList = ['naveen', 'shabnam','sonia','anshu','hem']; aList.sort();

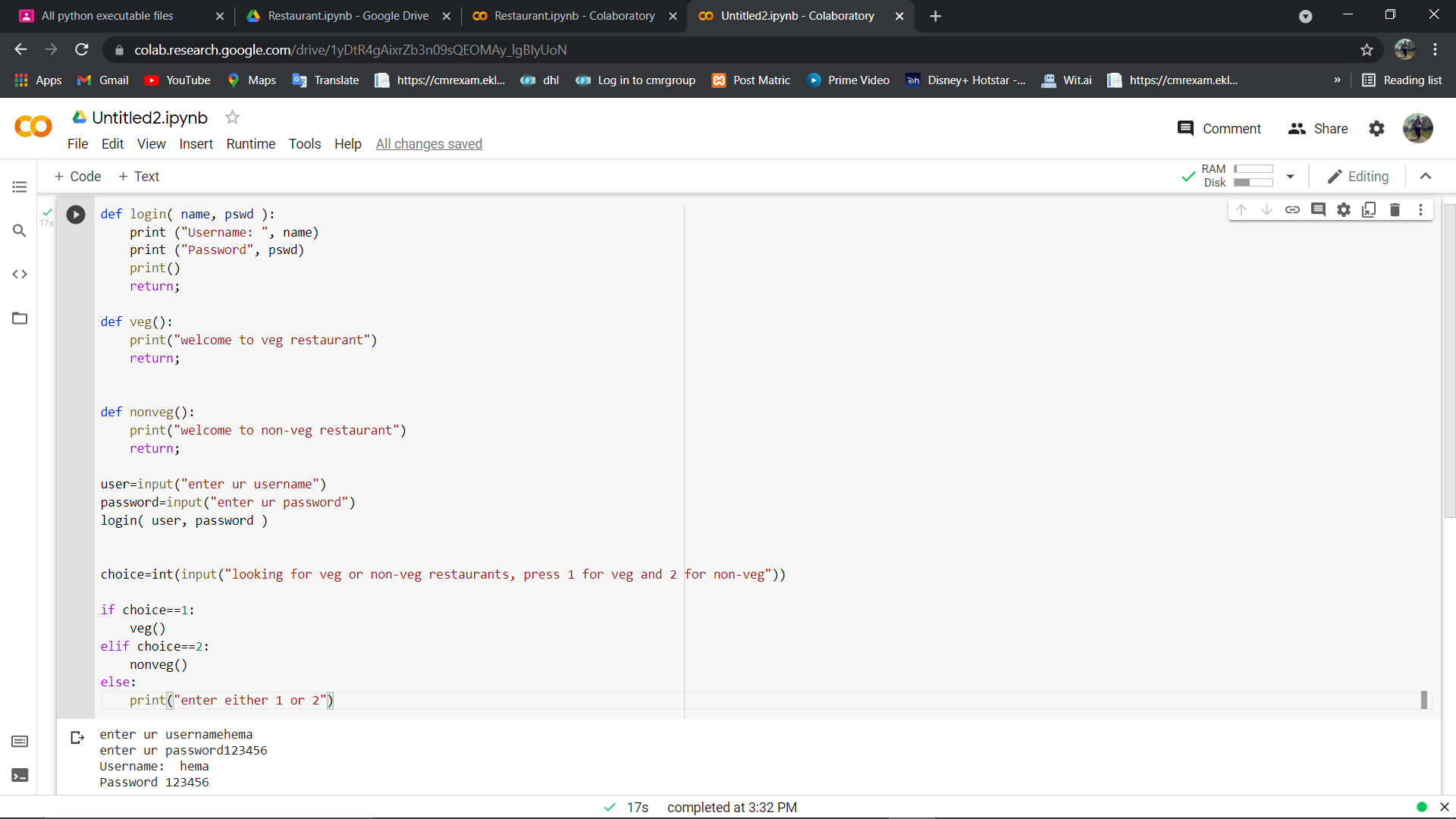
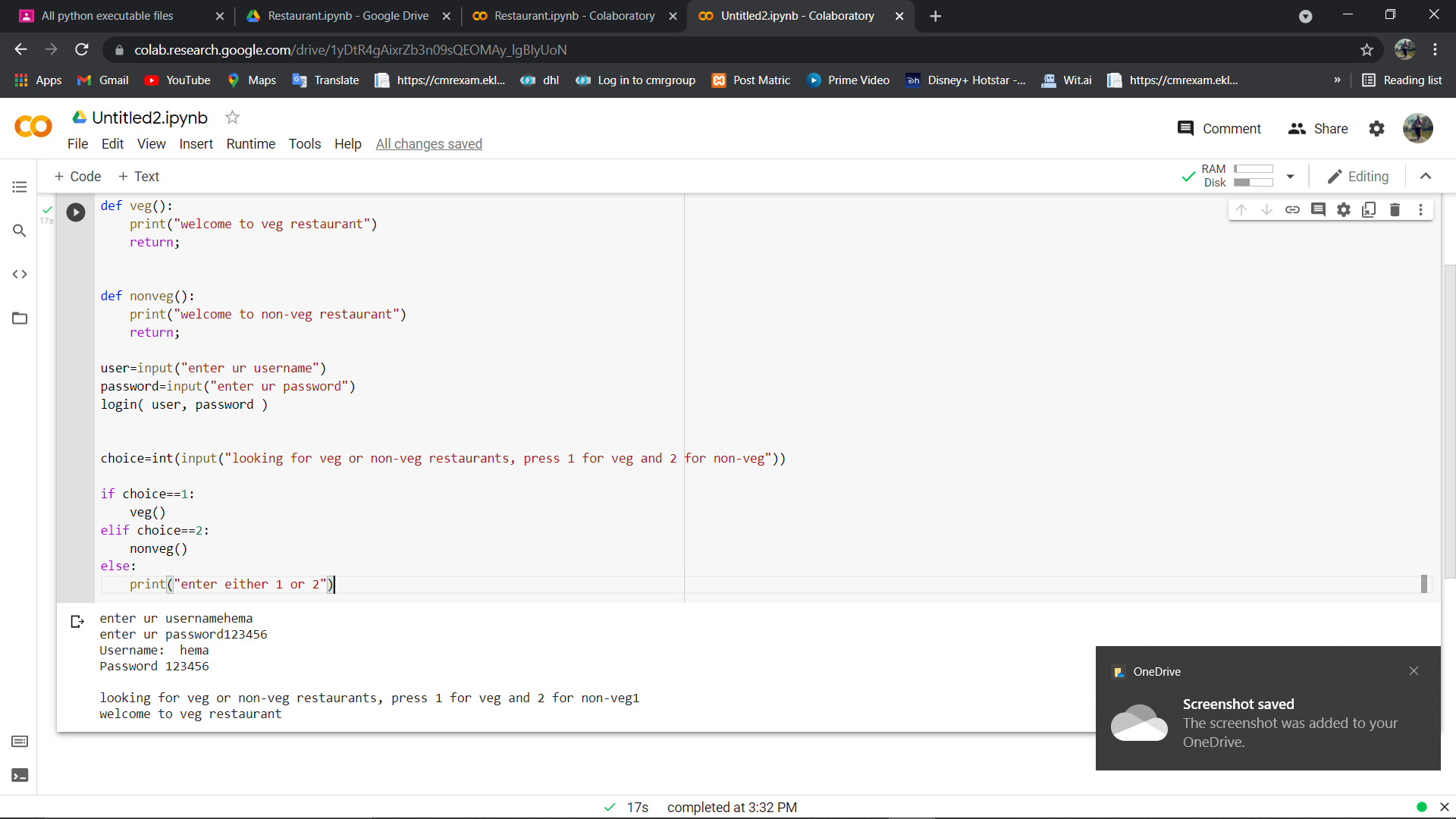
print ("List : ", aList)

List : ['anshu', 'naveen', 'hem', 'shabnam', 'sonia']

**RESTAURANT**

**Program 1**

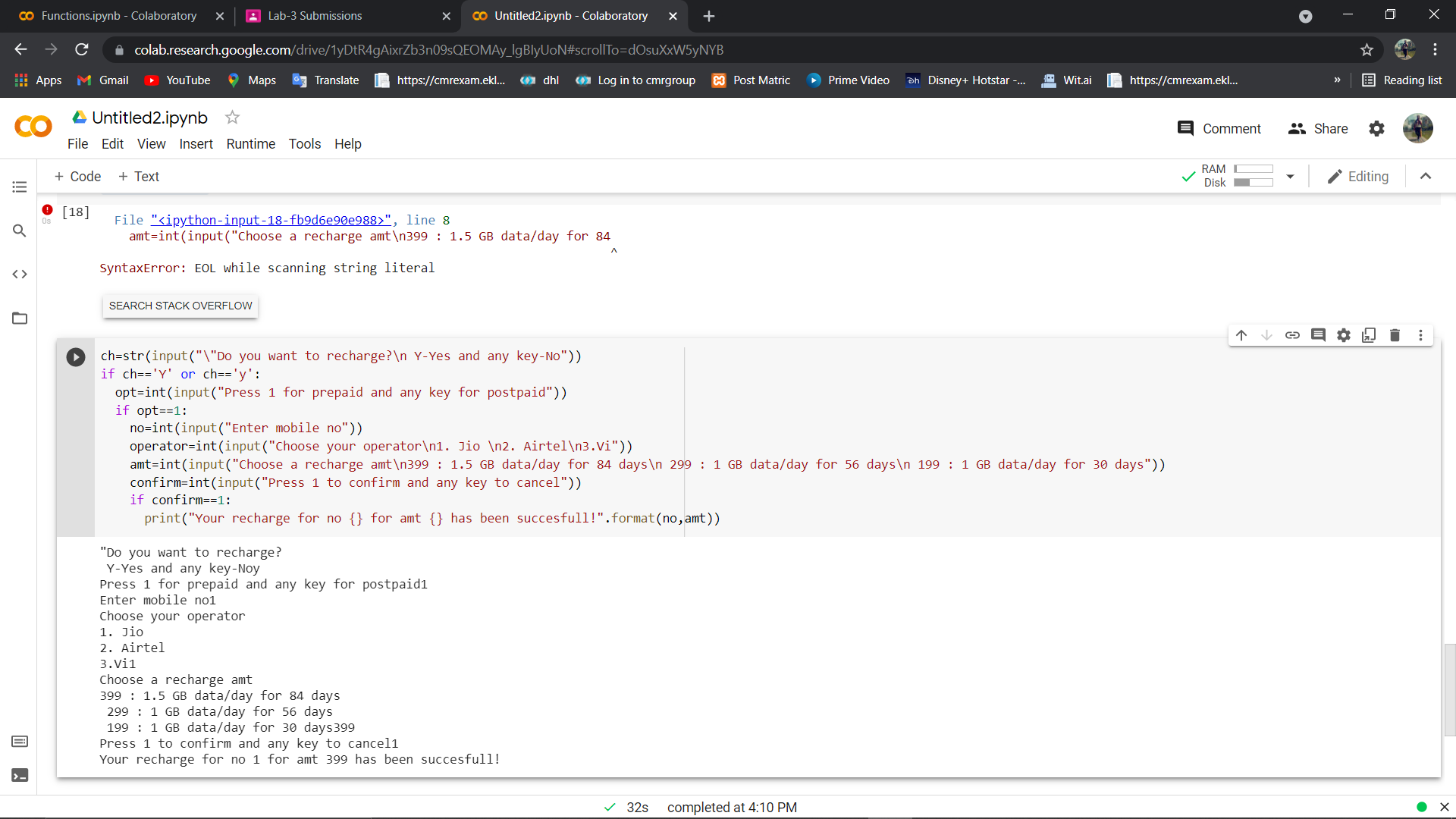
**Objective:** To check whether the user is looking for veg or non veg restaurant

**CODE & OUTPUT:**

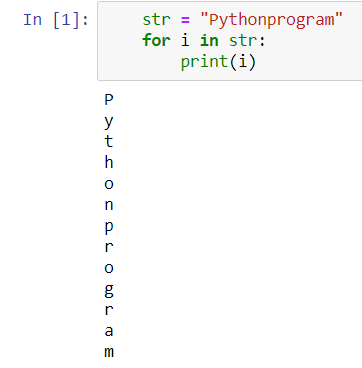
**RECHARGE**

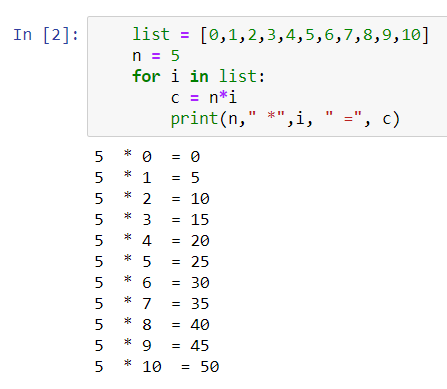
**Program 1**

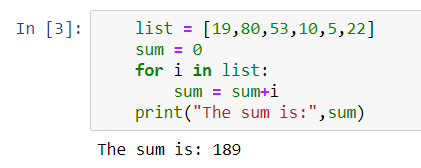
**Objective:** user can recharge to their given mobile numbers by selecting packages

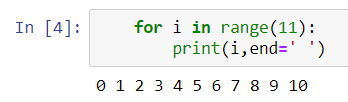
**CODE & OUTPUT**

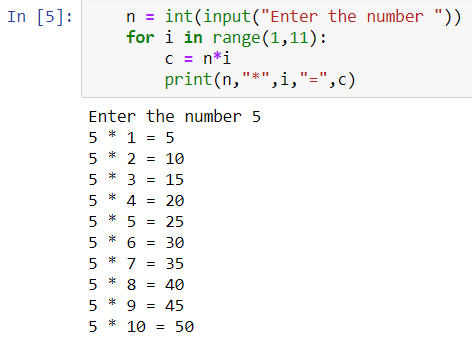
**Loops-Breaks—Continue Pass in Python**

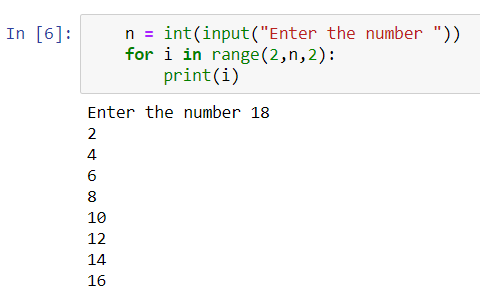
****

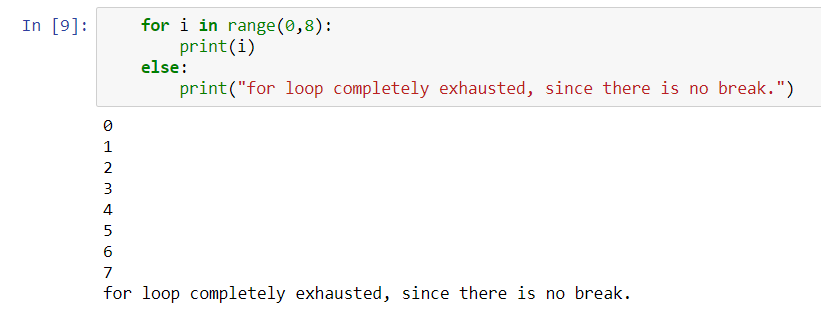
****

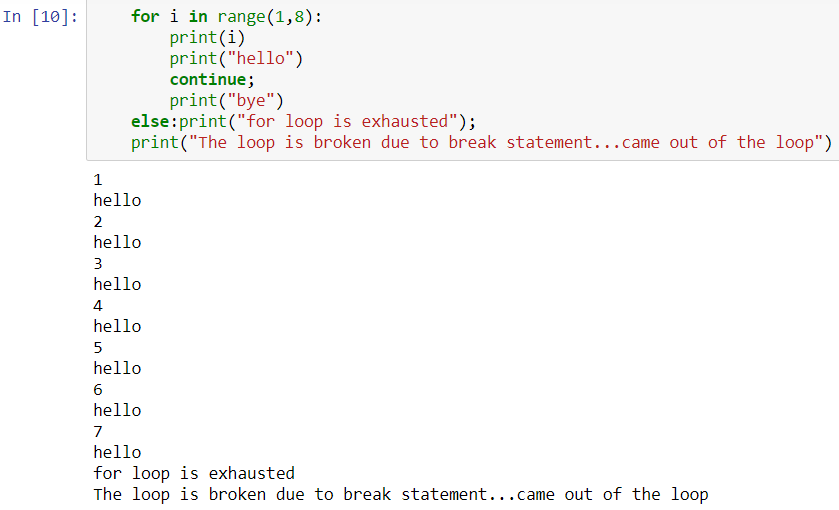
****

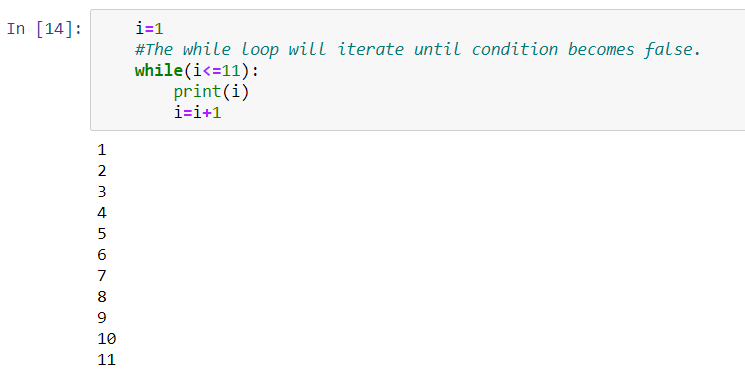
****

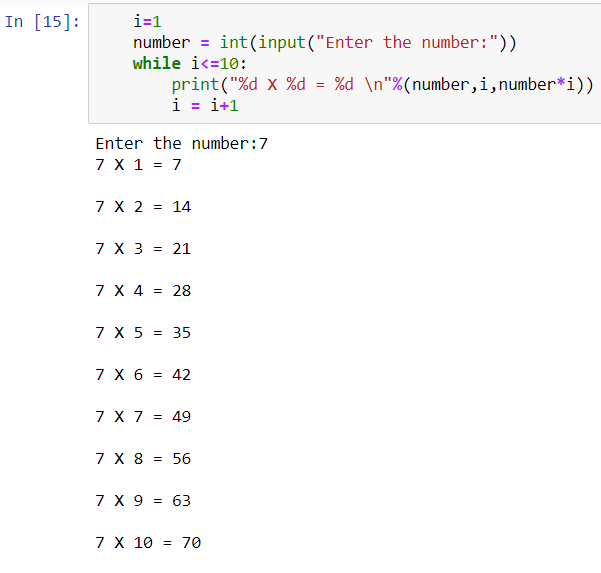
****

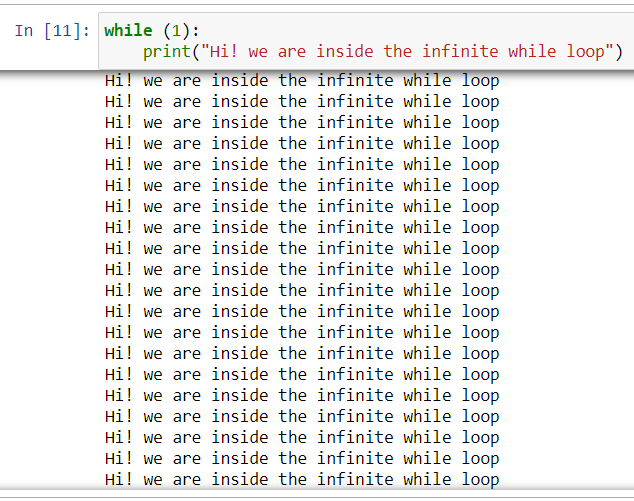
****

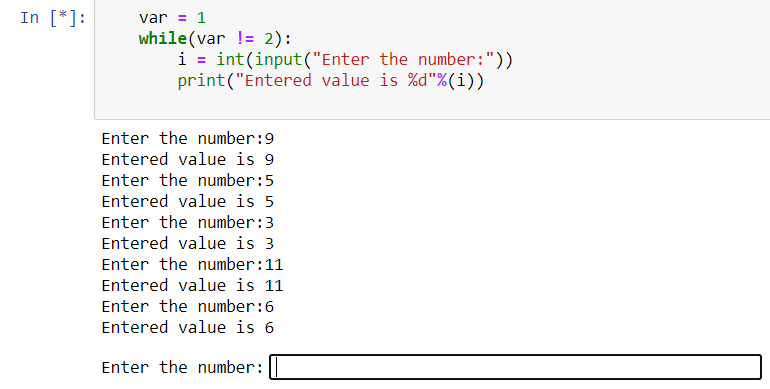
****

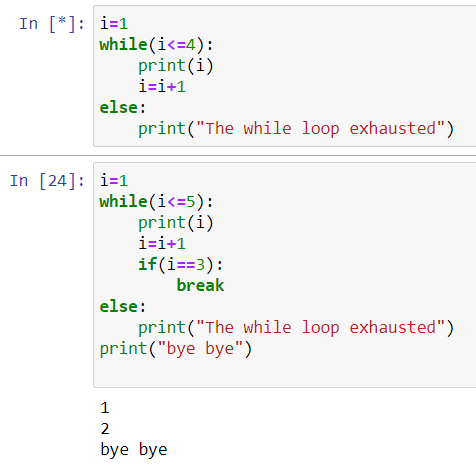
****

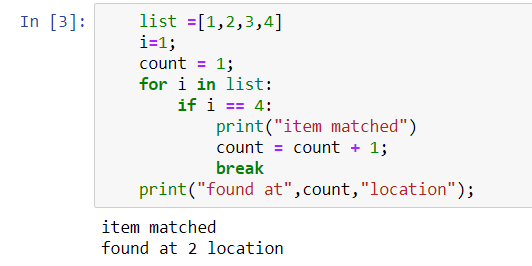
****

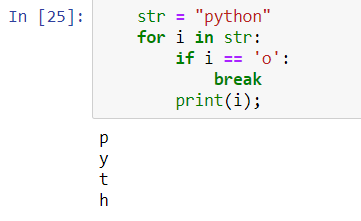
****

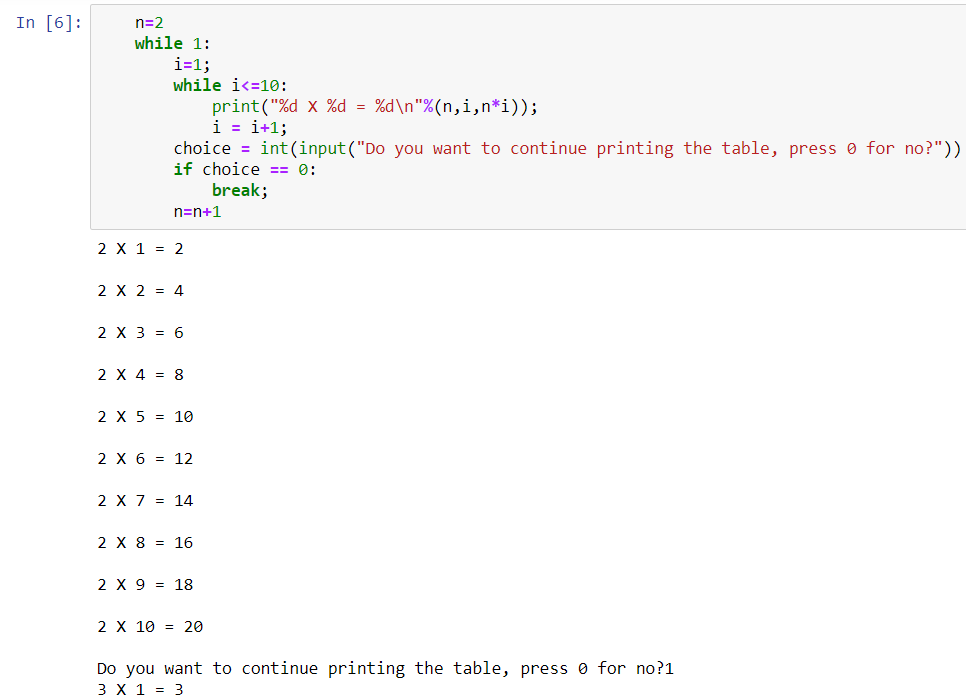
****

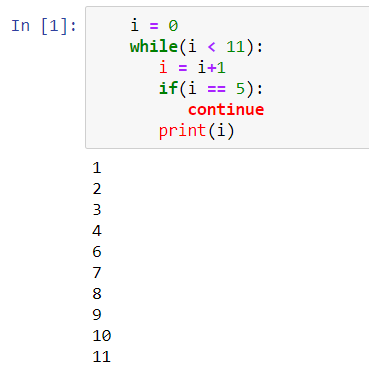
****

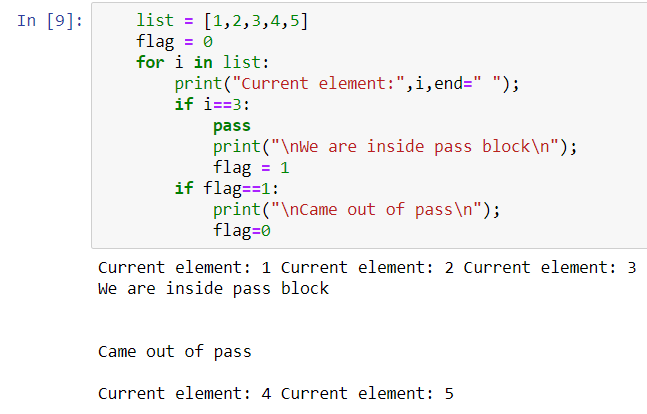
****

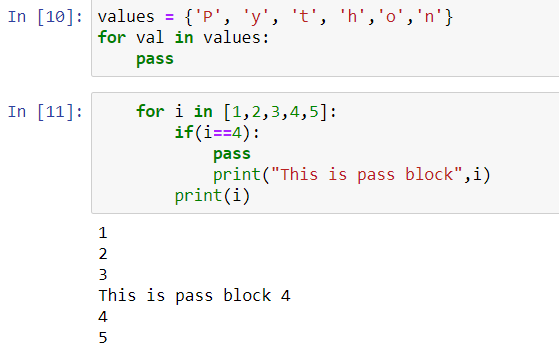
****

****

****

****

****

****

**Strings in Python**

**Program 1**

**Objective:** Code to print/combine the given two strings into one

**Code & Output:**

### var1 = 'Hello World!'

var2 = "Python Programming" print(var1," ",var2)

Hello World! Python Programming

**Program 2**

**Objective:** code to print elements present at specific index value

**Code & Output:**

### var1 = 'Hello World!'

var2 = "Python Programming"

print (var1[0])

print (var2[1:5])

H

ytho

**Program 3**

**Objective:** To change the specific string value in python

**Code & Output:**

### var1 = 'welcome home!'

print ("Updated String :- ", var1[0:8] + 'hem')

Updated String :- welcome hem

**Program 4**

**Objective:** To print the values present at given index location

**Code & Output:**

### str1 = input("Please Enter Your Own String : ")

str2 = str1 str3=str1[:] str4=str1[2:6]

print("The Final String :Str2=",str2) print("The Final String:Str3==",str3) print("The Final String:Str4==",str4)

Please Enter Your Own String : hello class The Final String : Str2 = hello class The Final String : Str3 = = hello class

The Final String : Str4 = = llo

**Program 5**

**Objective:** Capitalize function in python which capitalizes only its first character.

**Code & Output:**

str = "this is string example. wow!!!";

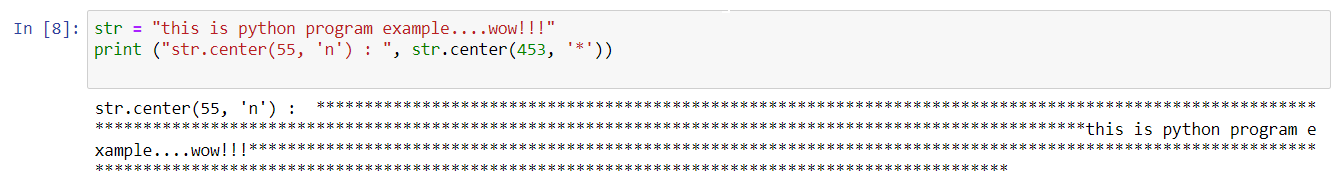
print ("str.capitalize() : ", str.capitalize())

str.capitalize() : This is string example. wow!!!

**Program 6**

**Objective:** To return centered in a string of length width. Padding is done using the specified fillchar. Default filler is a space.

**Code & Output:**



**Program 7**

**Objective:** To return the number of occurrences of substring sub in the range [start, end]

**Code & Output:**

str = "this is string example. wow!!;

sub = "i";

print ("str.count(sub, 4, 40) : ", str.count(sub,4,40)) sub = "wow";

print ("str.count(sub) : ", str.count(sub))

str.count(sub, 4, 40) : 2

str.count(sub) : 1

**Program 8**

**Objective:** To determine if string str occurs in string, or in a substring of string if starting index beg and ending index end are given.

**Code & Output:**

str1 = "this is string example…wow!!!";

str2 = "is";

|  |  |  |
| --- | --- | --- |
| print | (str1.find(str2)) |  |
| print | (str1.find(str2, | 10)) |
| print | (str1.find(str2, | 40)) |
| 2 |  | |
| -1 |
| -1 |

**Program 9**

**Objective:** To checks whether the string consists of alphanumeric characters.

**Code & Output:**

str = "this2009"; # No space in this string print (str.isalnum())

str = "this is string example. wow!!!";

print (str.isalnum())

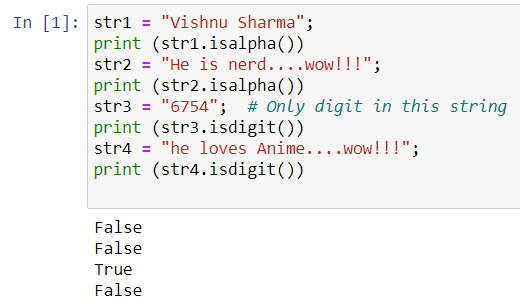
True

False

**Program 10**

**Objective:** To check whether string consist of alphabets and numbers without spaces using isalpha and isdigit.

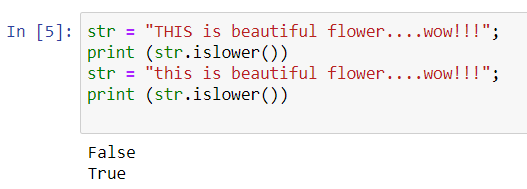
**Code & Output:**



**Program 11**

**Objective:** To check whether the string first letter is a small letter

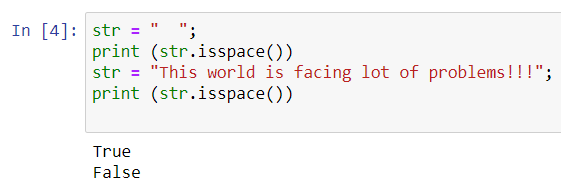
**Code & Output:**



**Program 12**

**Objective:** To check whether the string is empty with only space

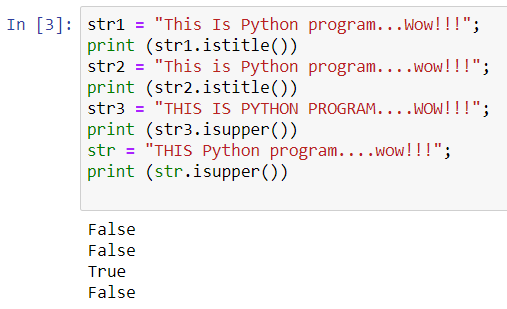
**Code & Output:**



**Program 13**

**Objective:** To check whether strings every first character after space is capital using istitle and to check if all character in string is capitalized using isupper

**Code & Output:**



**Program 14**

**Objective:** To use join() function which returns a string in which the string elements of sequence have been joined by str separator.

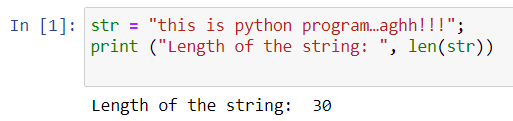
**Code & Output:**



**Program 15**

**Objective:** To check for the length of string

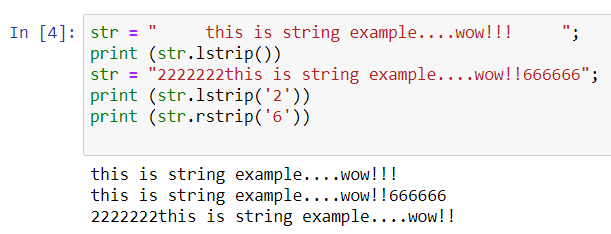
**Code & Output:**



**Program 16**

**Objective:** To use istrip() function which returns a copy of the string in which all chars have been stripped from the beginning of the string (default whitespace characters)

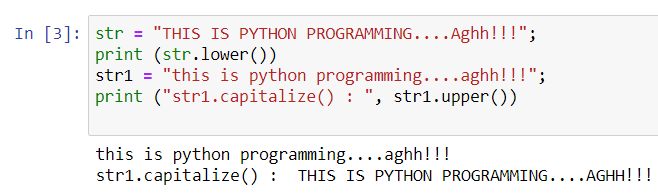
**Code & Output:**



**Program 17**

**Objective:** To use lower () and capitalize function which convrts the string to lower and upper cases respectively

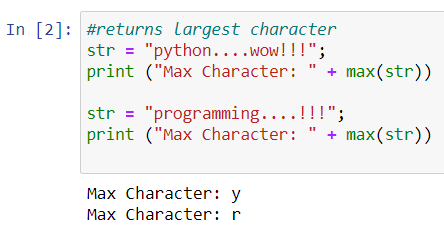
**Code & Output:**



**Program 18**

**Objective:** To use max() function which returns largest character

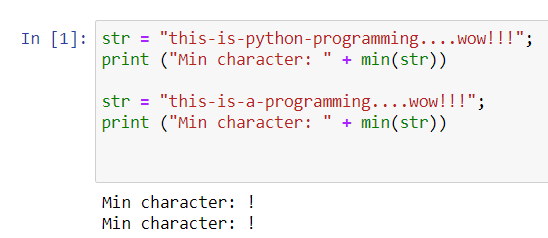
**Code & Output:**



**Program 19**

**Objective:** To use min() function which returns smallest character

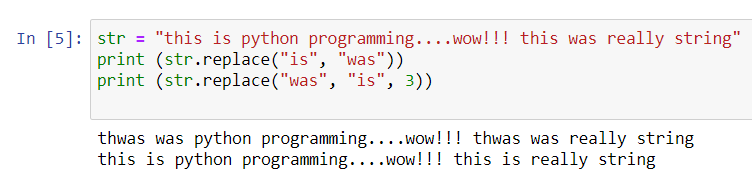
**Code & Output:**



**Program 20**

**Objective:** To replace a certain character in string with another character

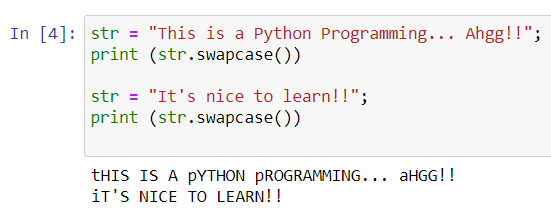
**Code & Output:**



**Program 21**

**Objective:** To swap the two strings using swapcase() function

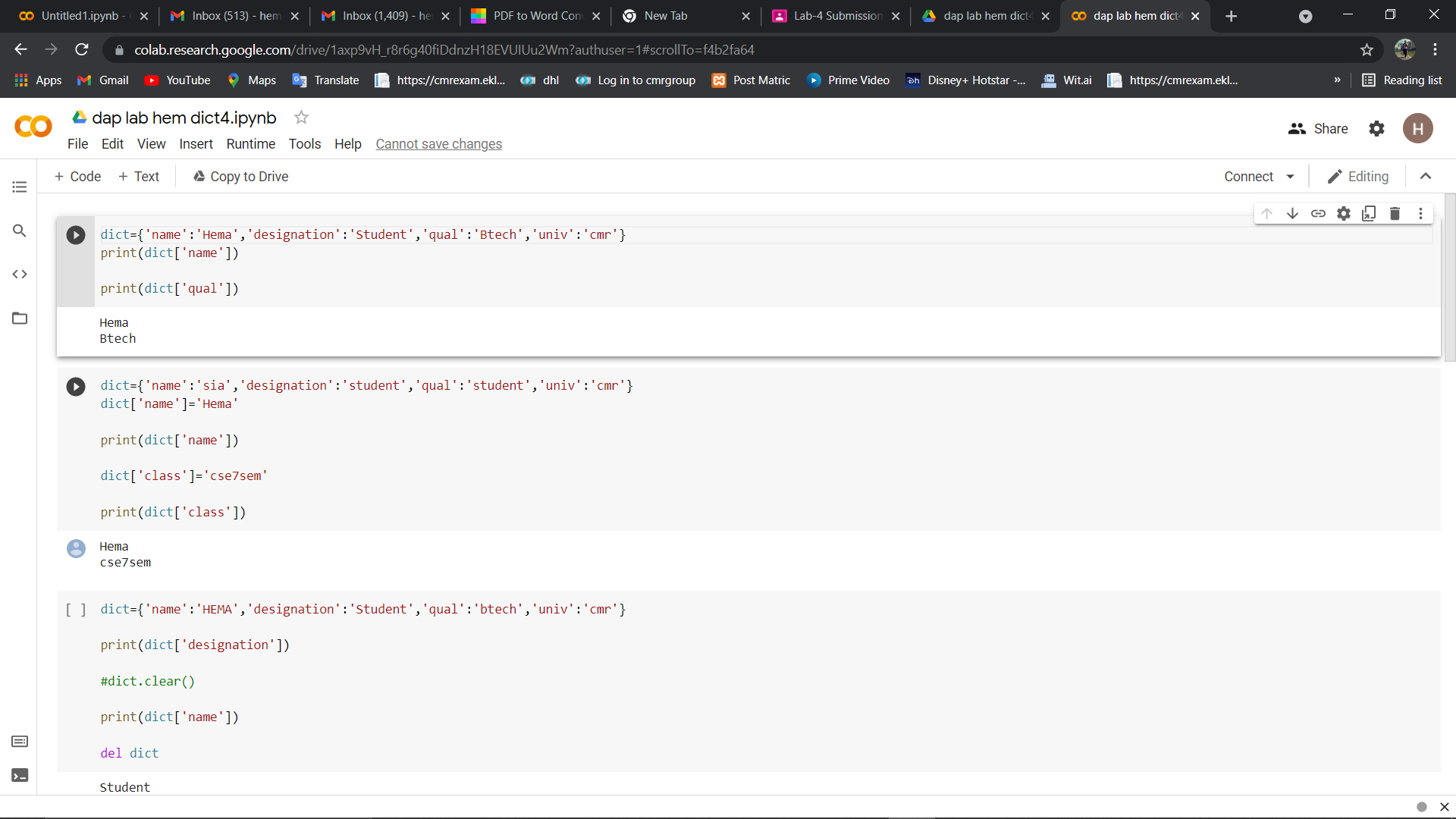
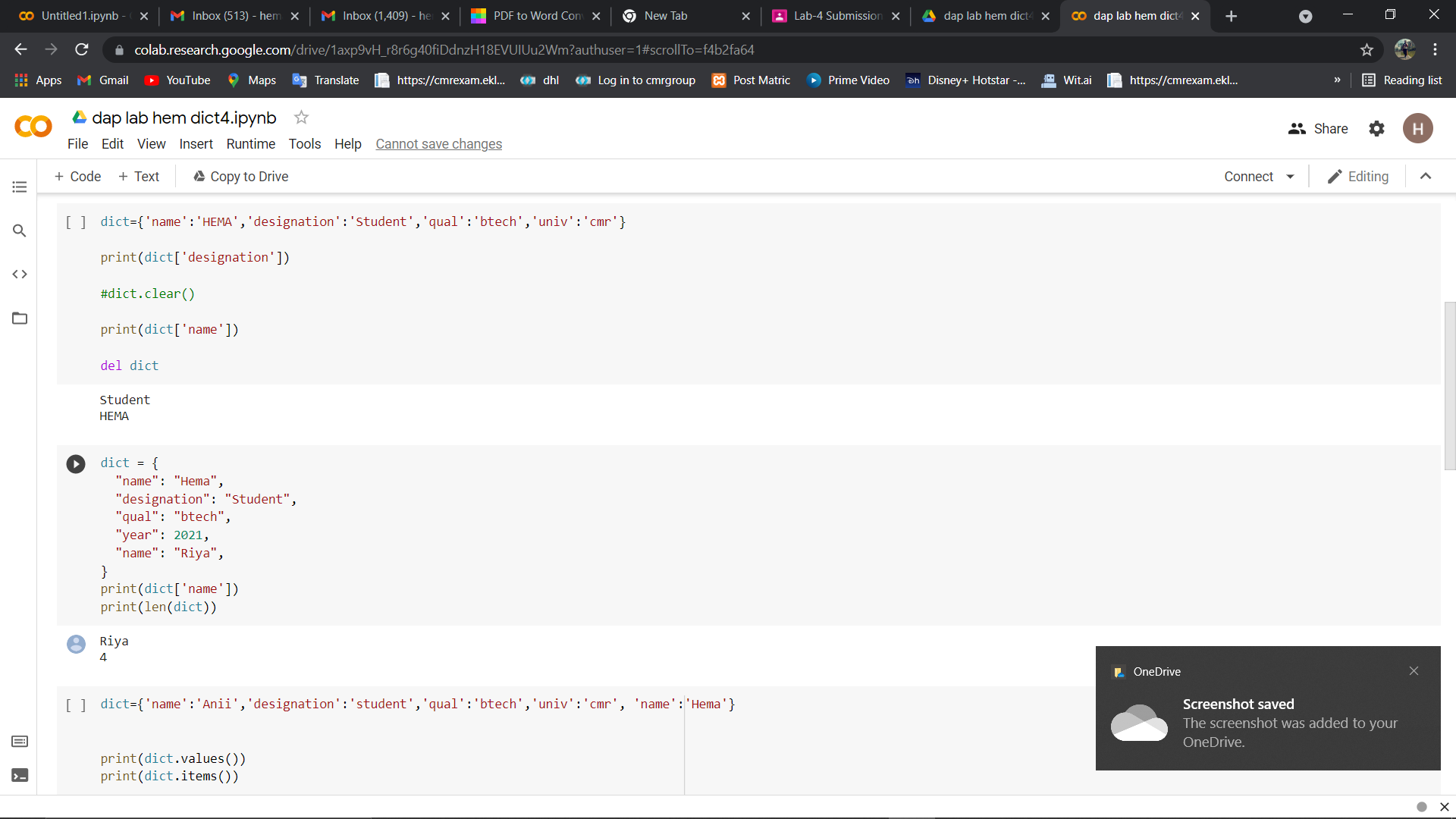
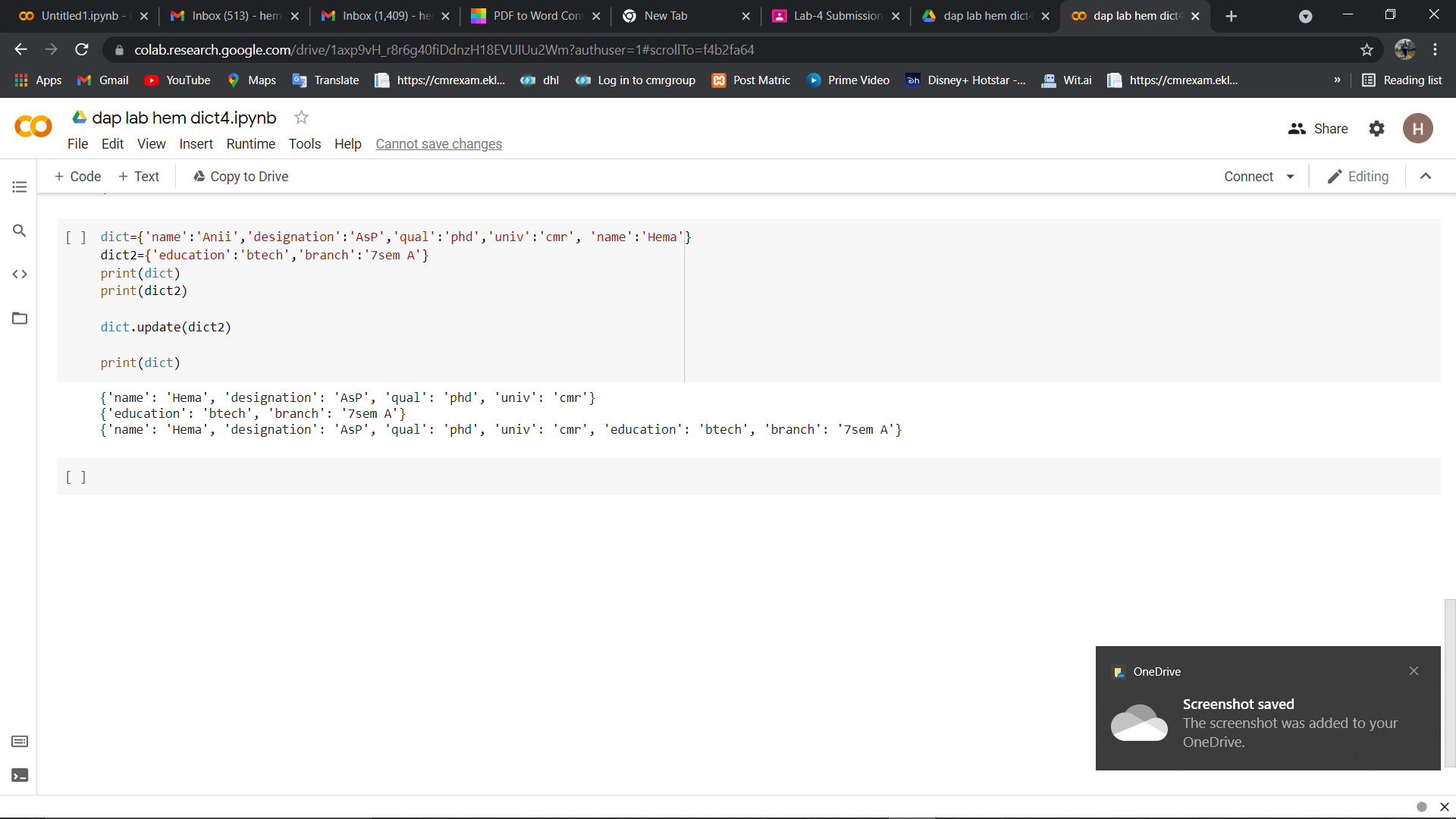
**Code & Output:**

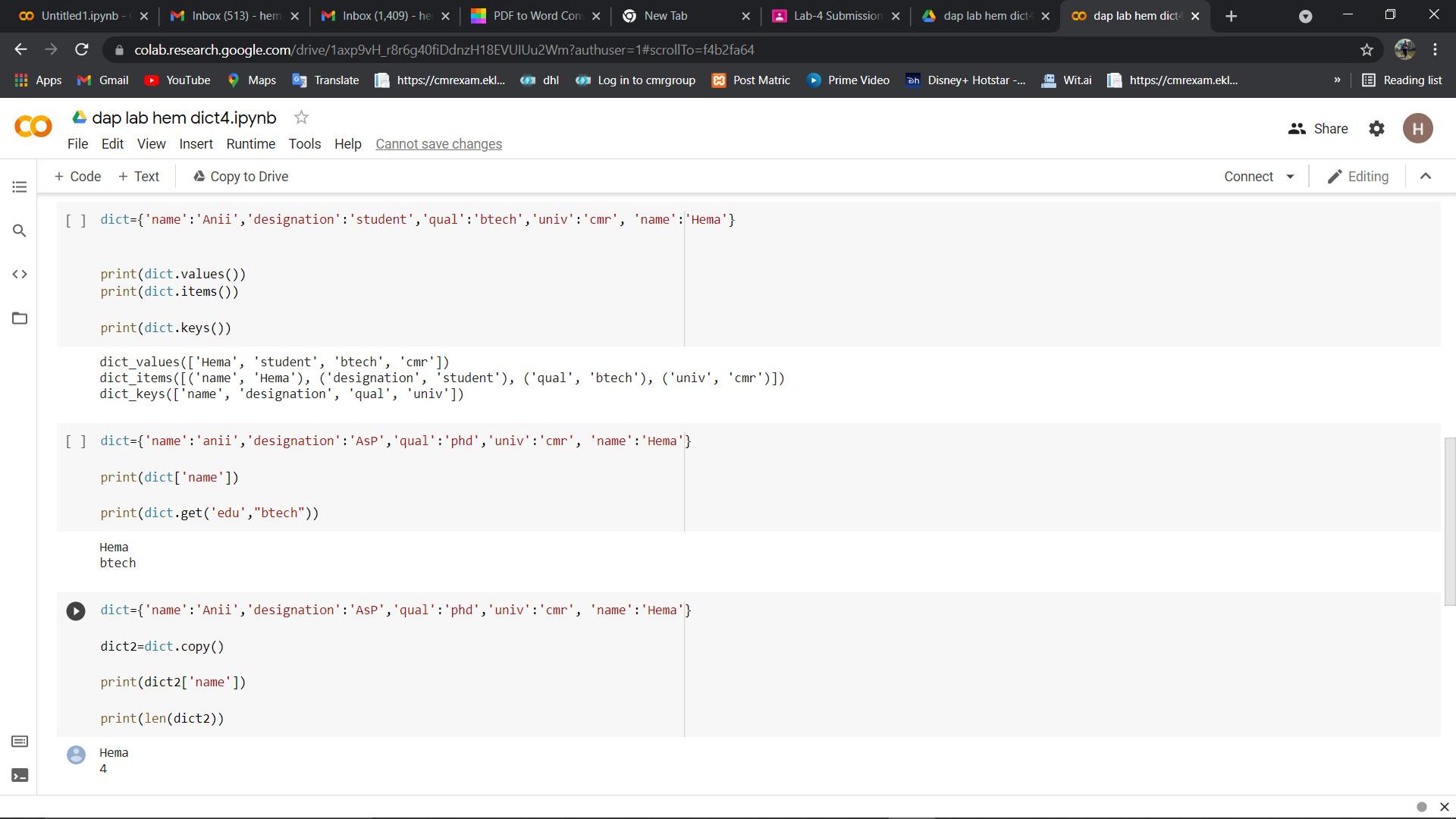


**Dictionaries in Python**

**Objective:** Dictionary values have no restrictions. They can be any arbitrary Python object, either standard objects or user-defined objects. Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples. In order to access the items of a dictionary refer to its key name. Key can be used inside square brackets. updating a dictionary by adding a new entry or a key-value pair, modifying an existing entry.

****

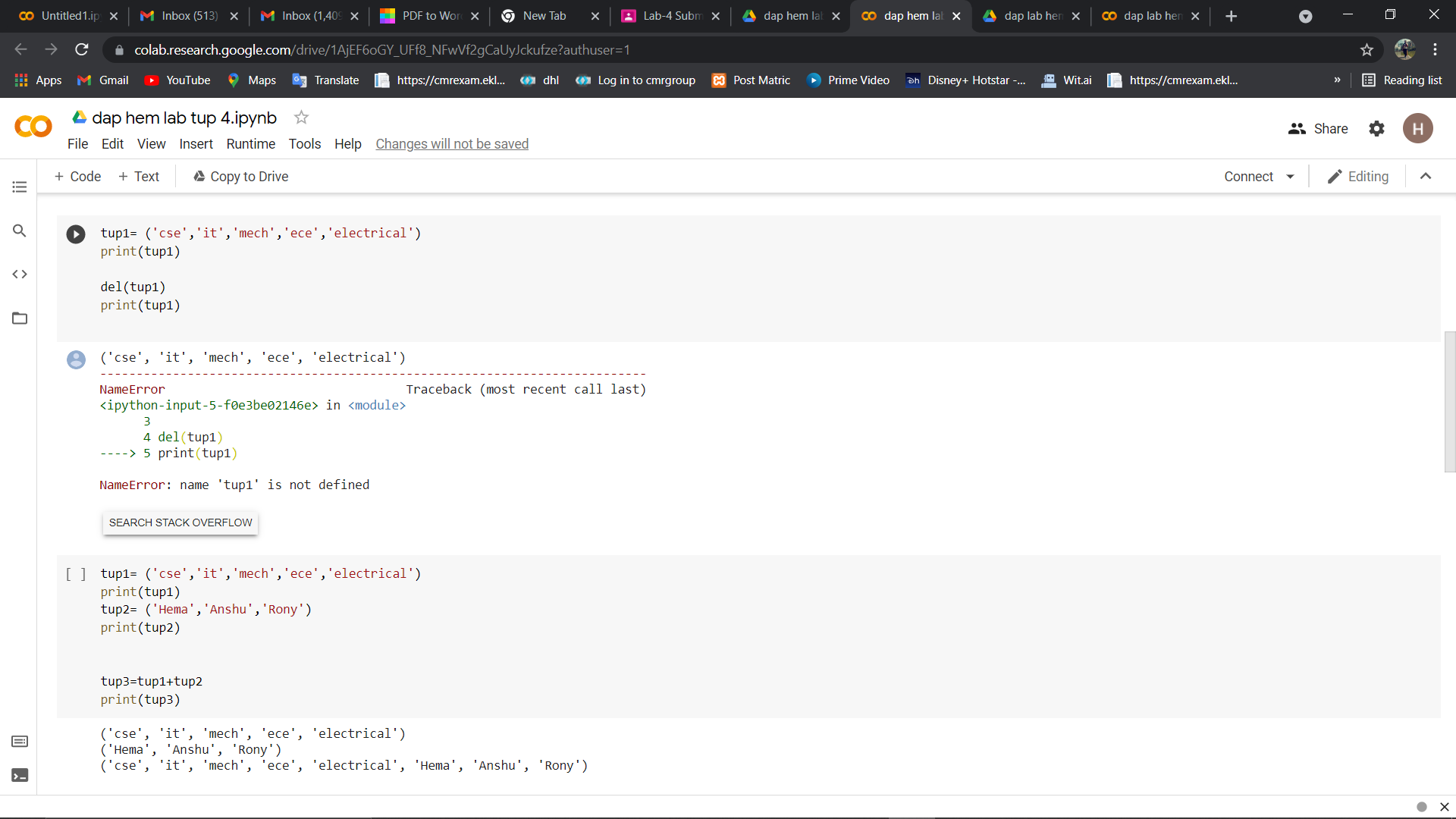
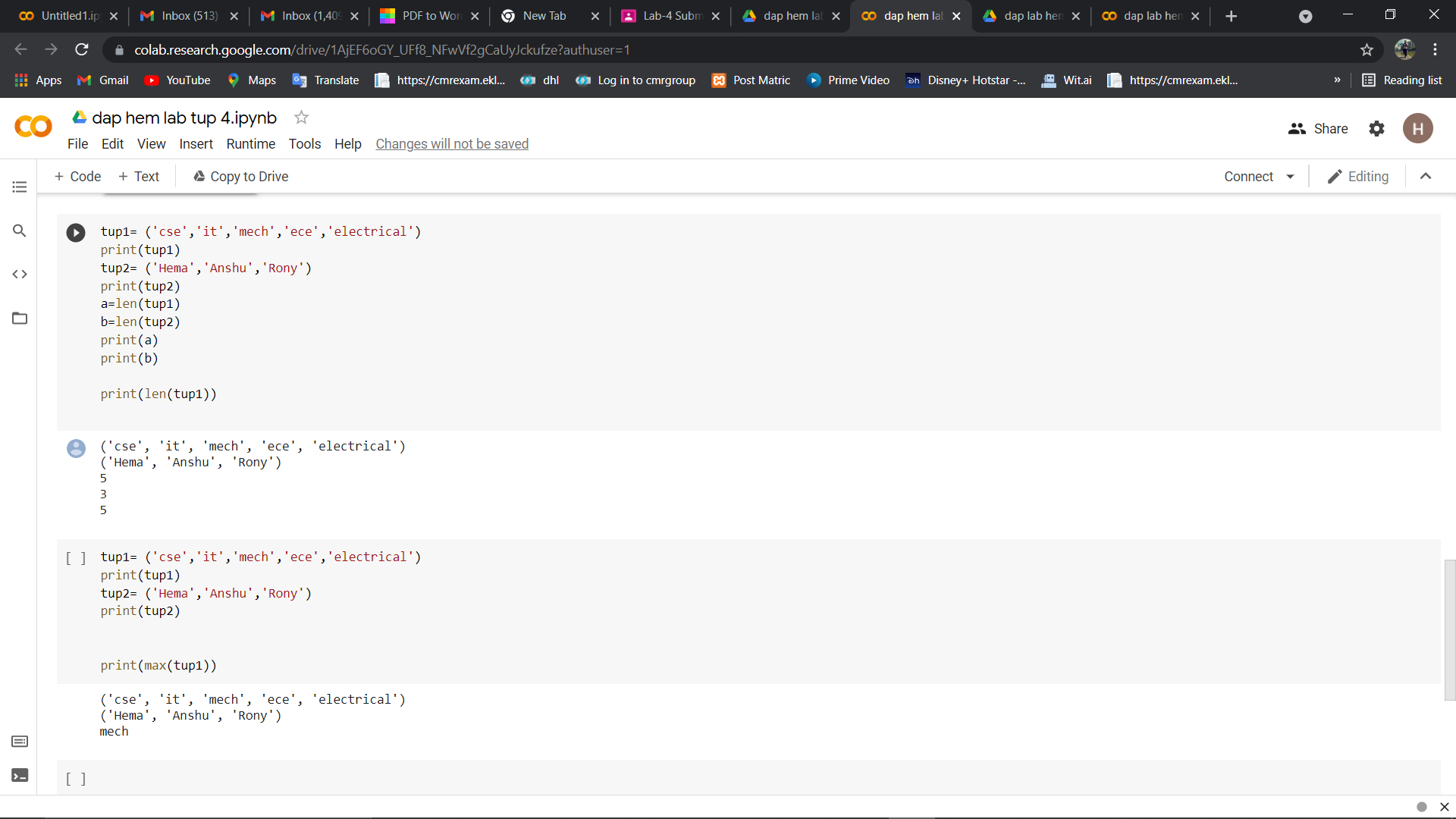
****

**TUPLES in Python**

**Objective:** Tuples are a core data structure in [Python](https://careerkarma.com/blog/what-python-is-used-for/). They let you store an ordered [sequence](https://careerkarma.com/blog/fibonacci-sequence-python/) of items. Tuples are immutable, ordered lists of data, unlike lists. Lists are mutable, which means you can change the contents of a list. Individual values in a tuple are called *items.* Tuples can store any data type.

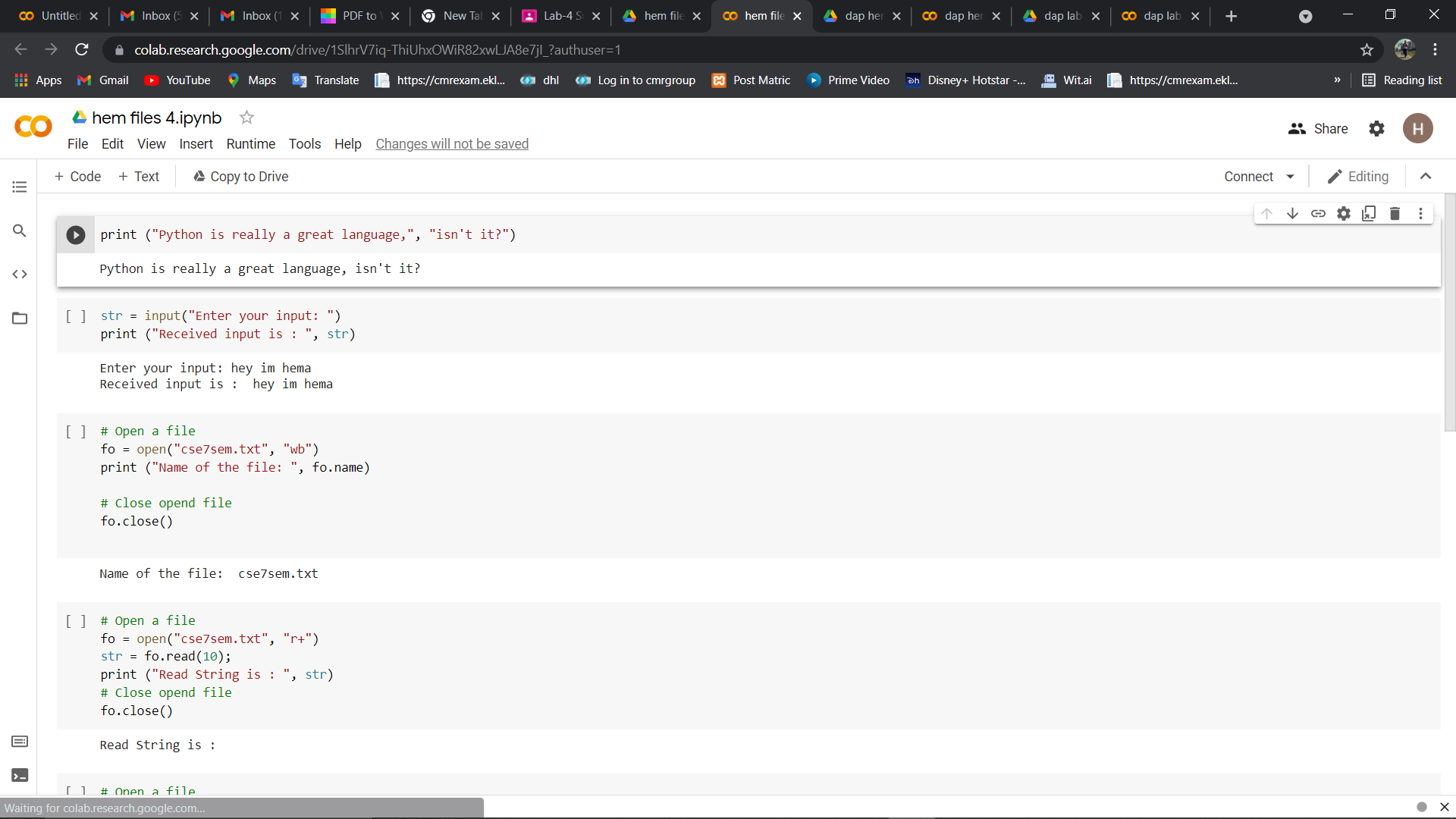
A tuple is a comma-separated sequence of items. This sequence is surrounded by parenthesis *(())*.

****

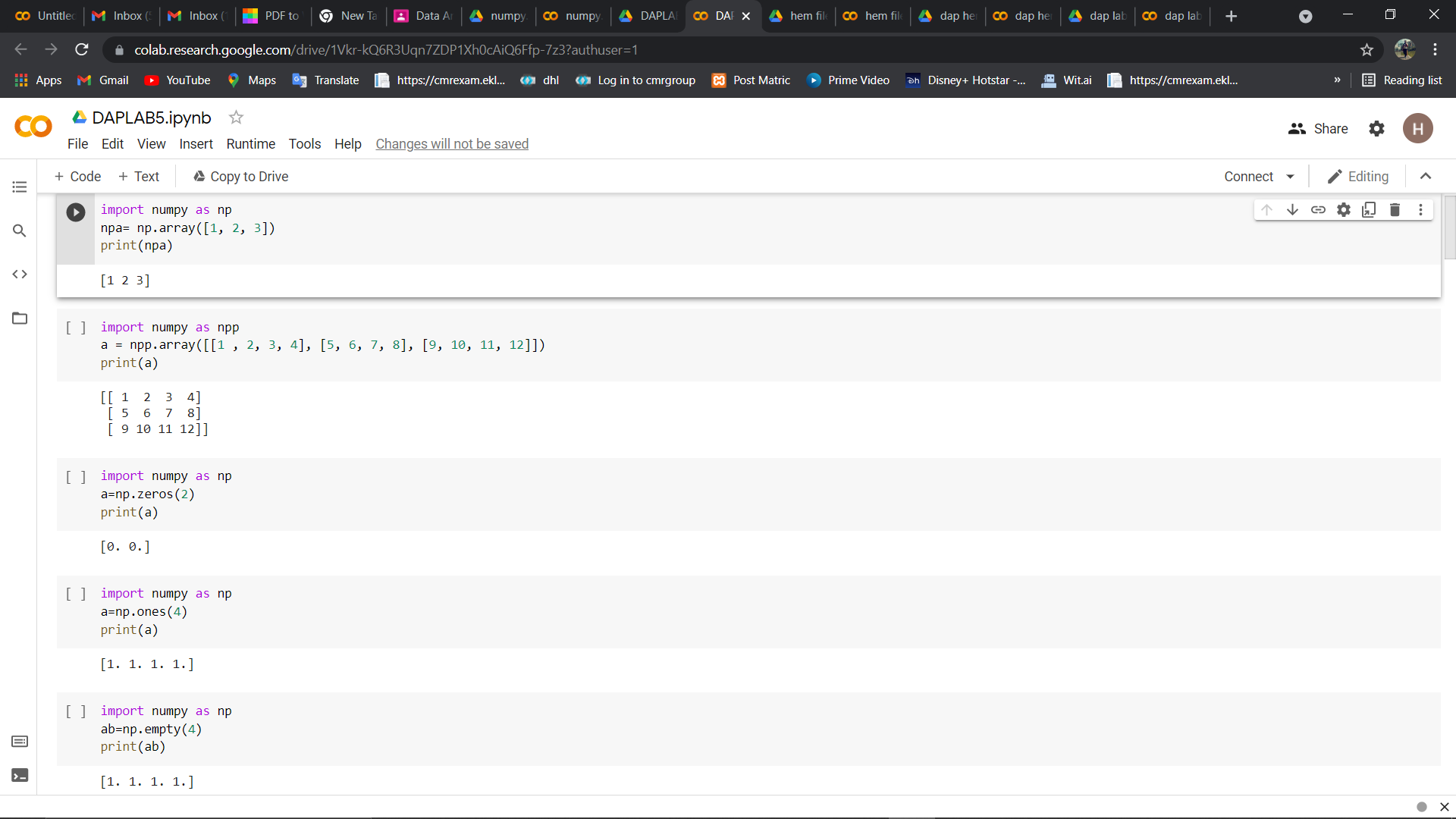
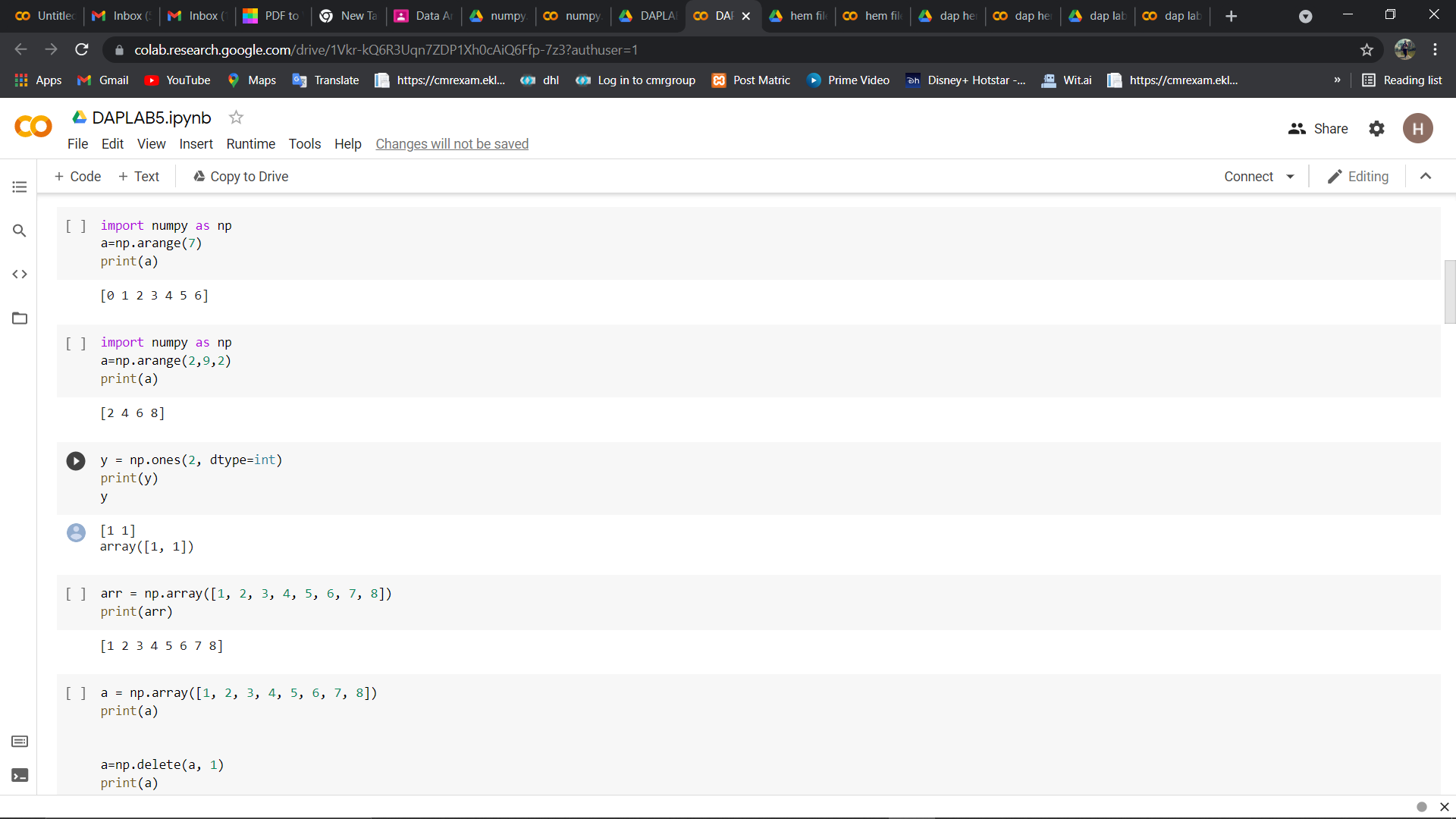
****

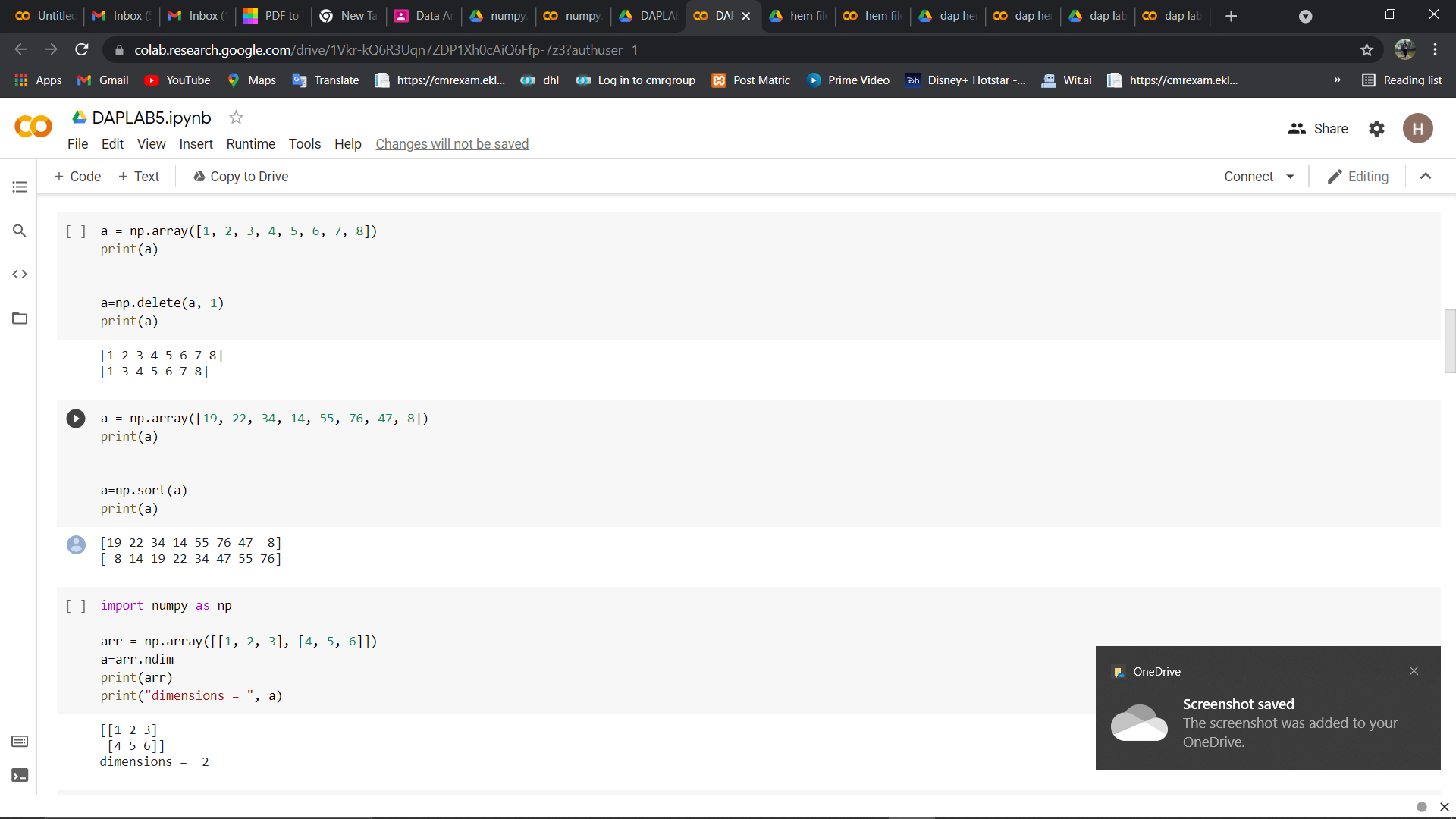
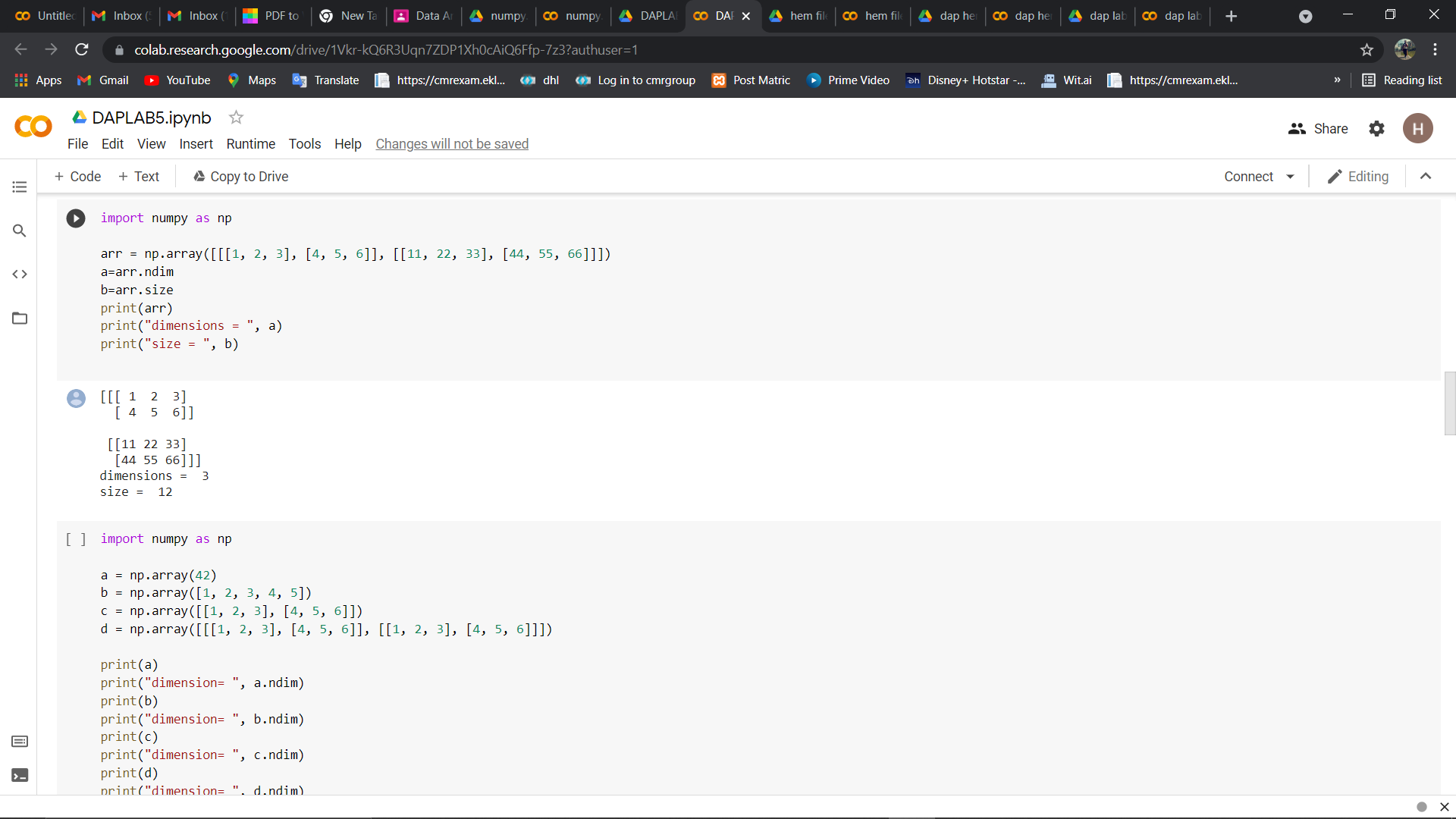
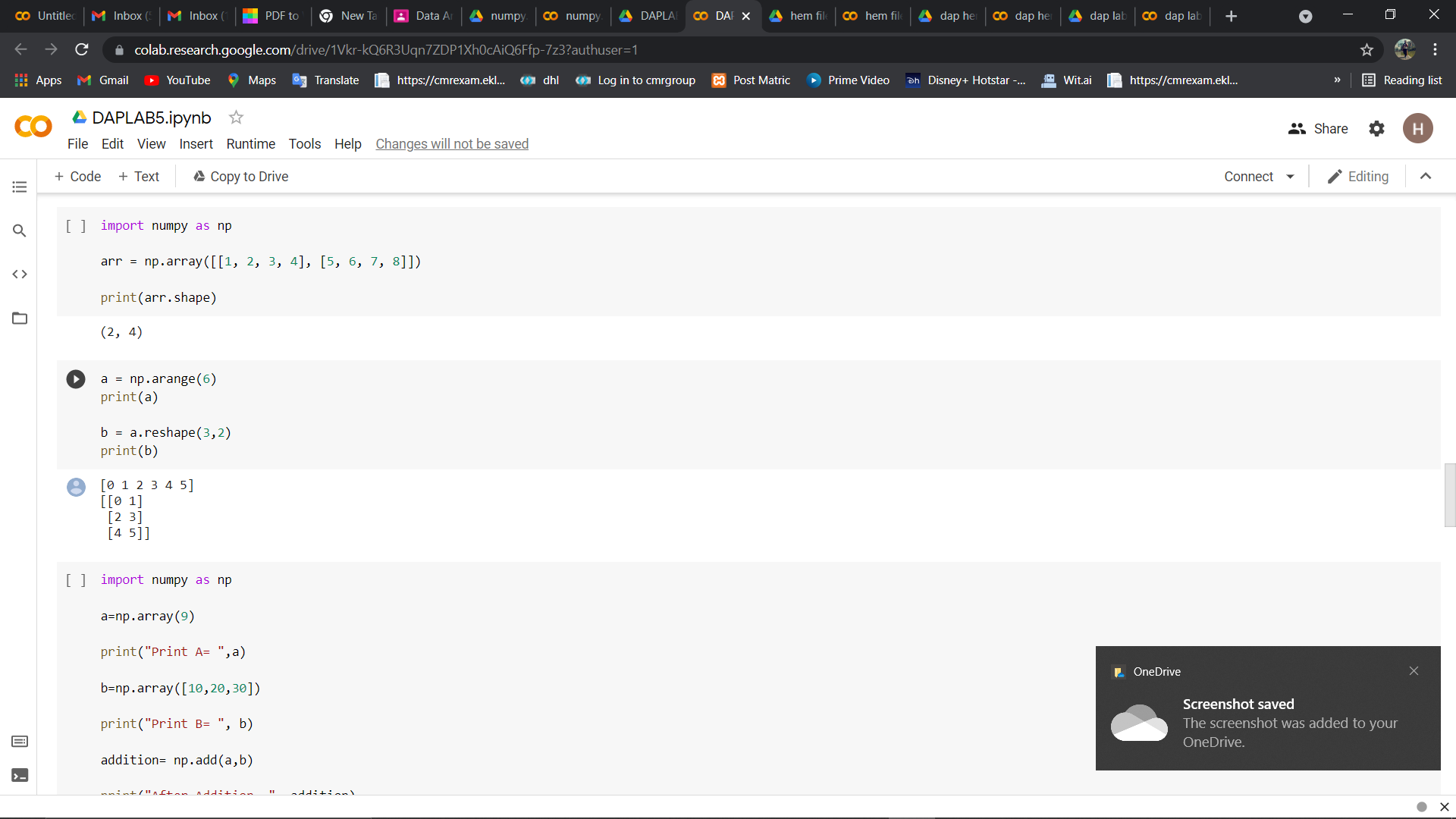
**Files in Python**

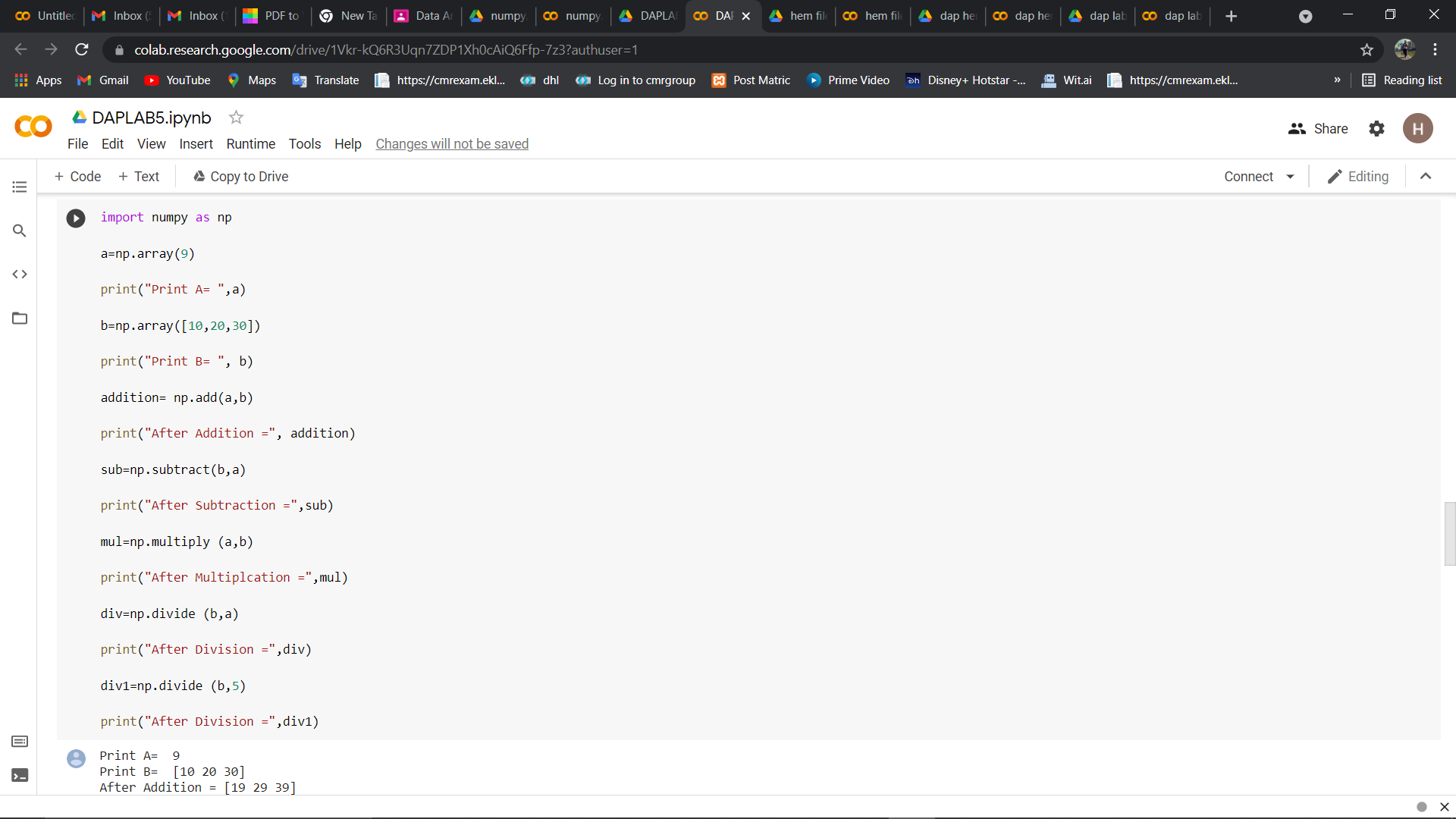
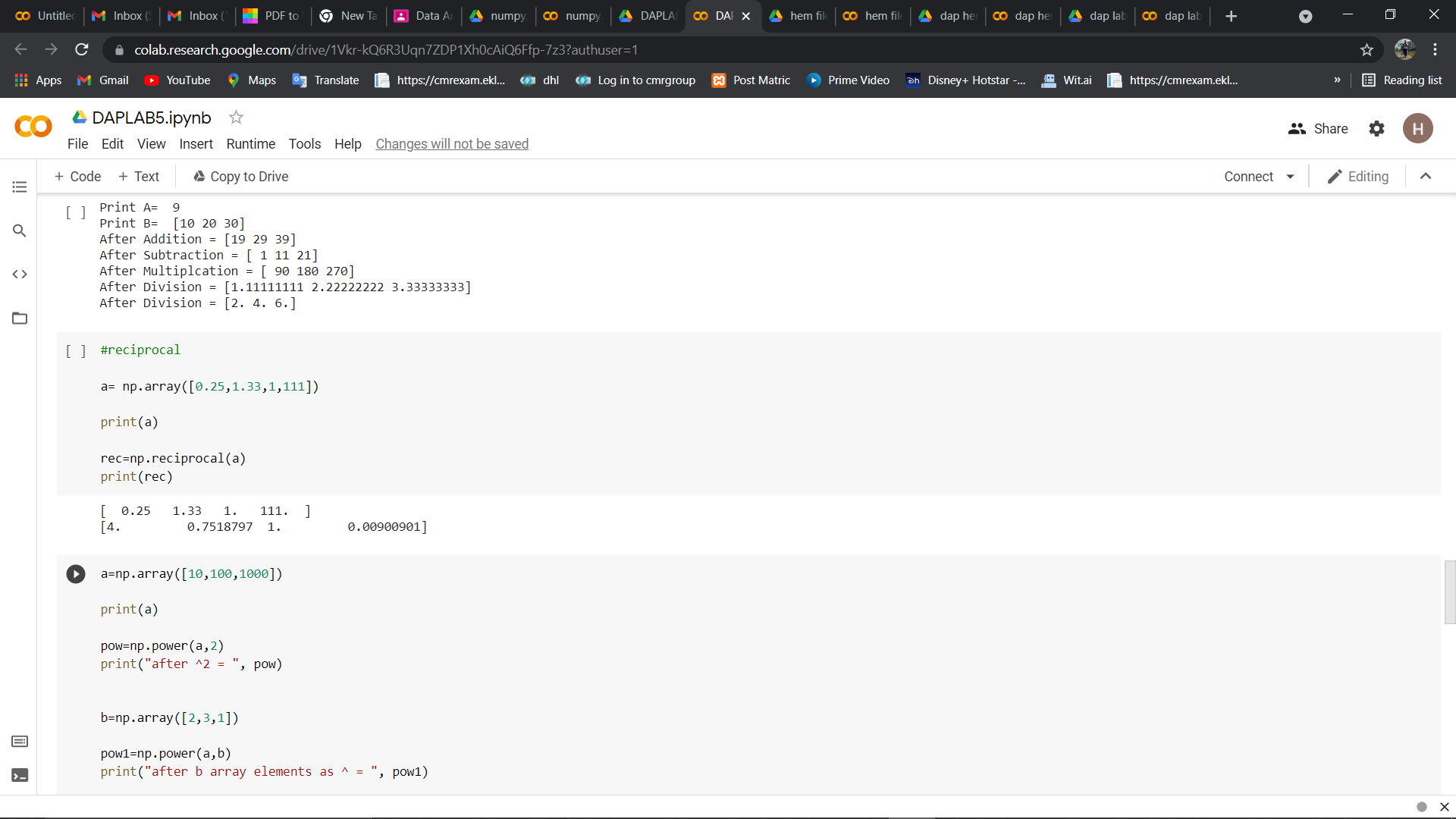
**Objective:** Python has a built-in open() function to open a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly. Closing a file will free up the resources that were tied with the file. It is done using the close() method available in Python.

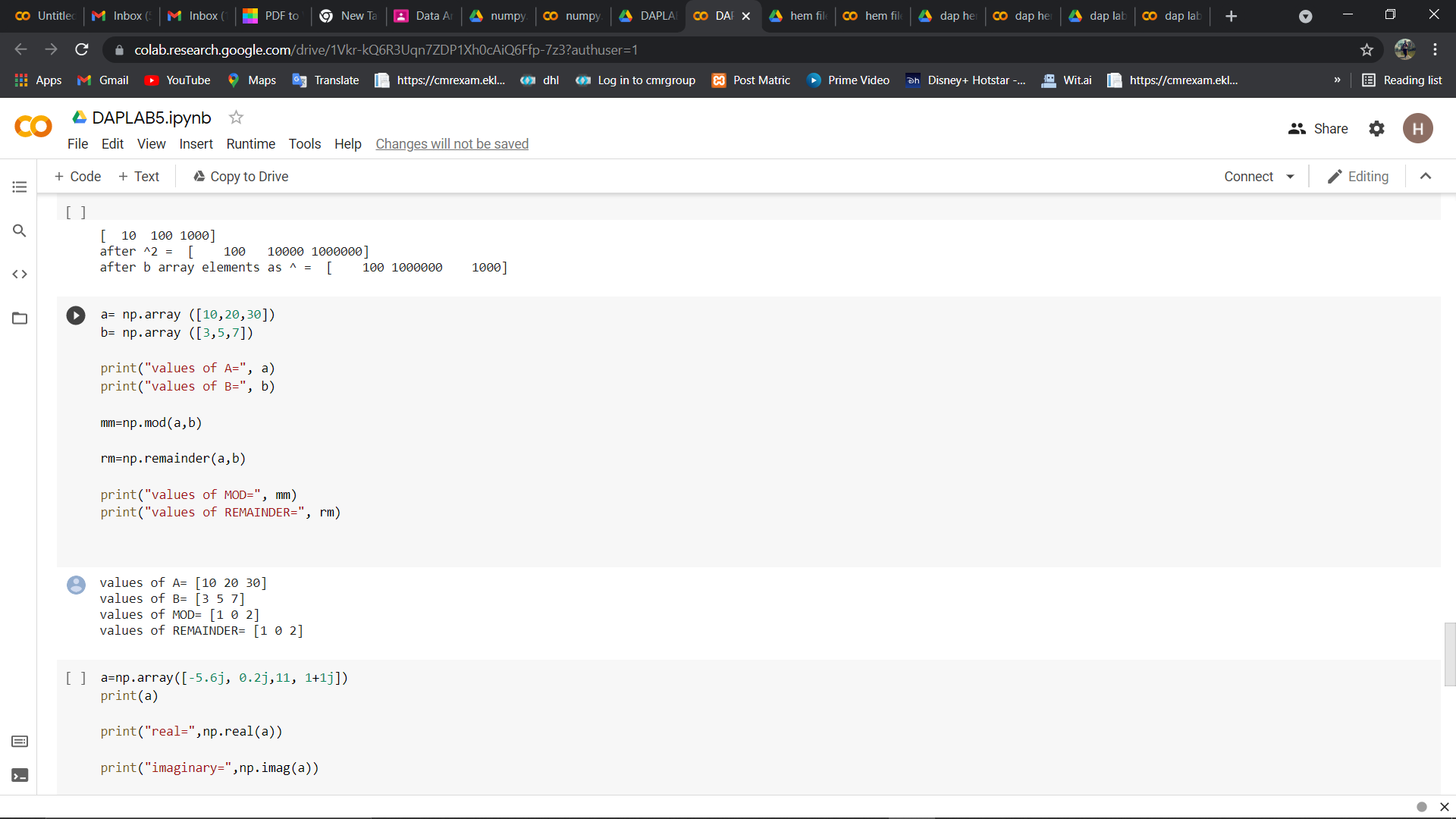
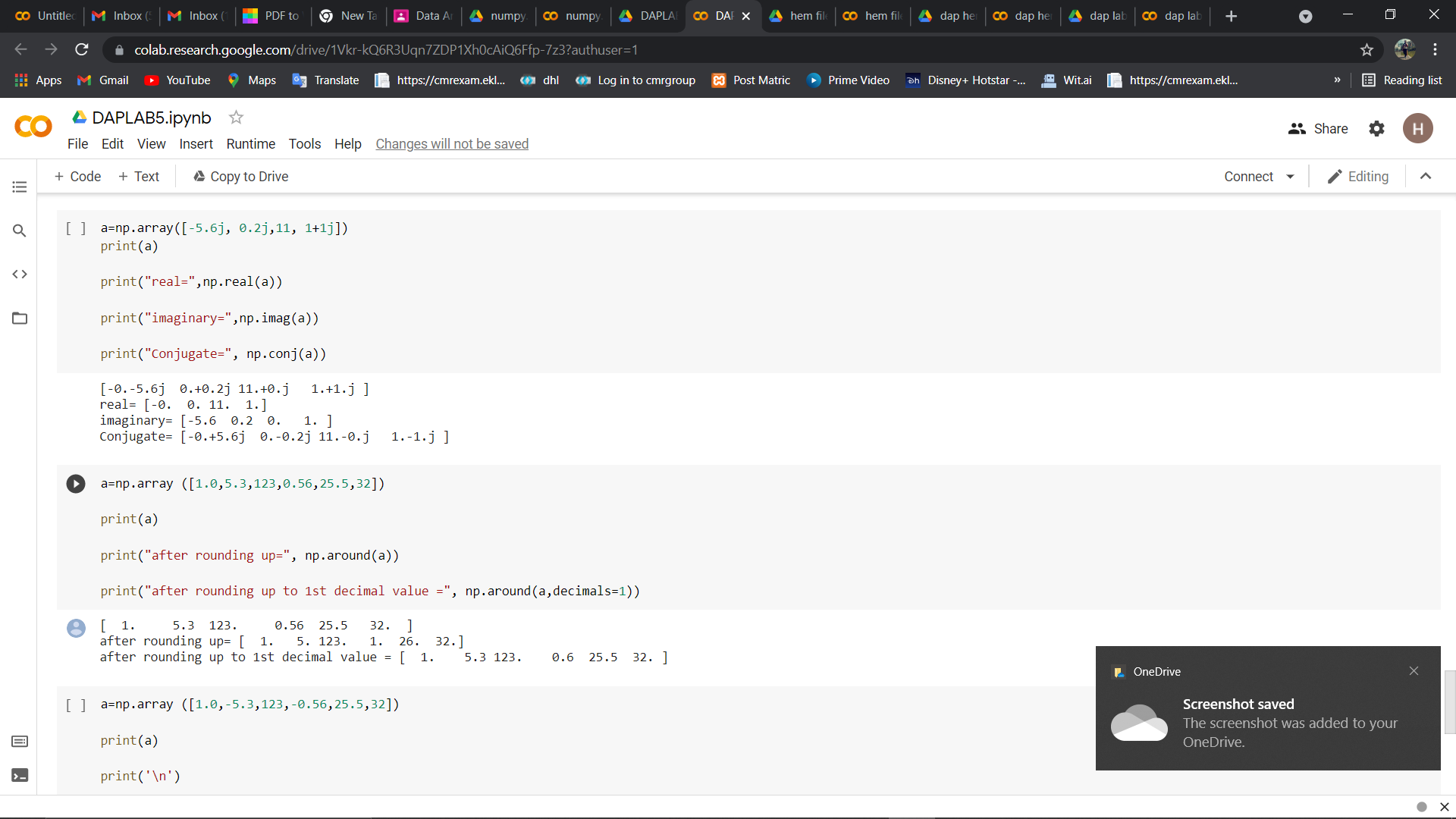
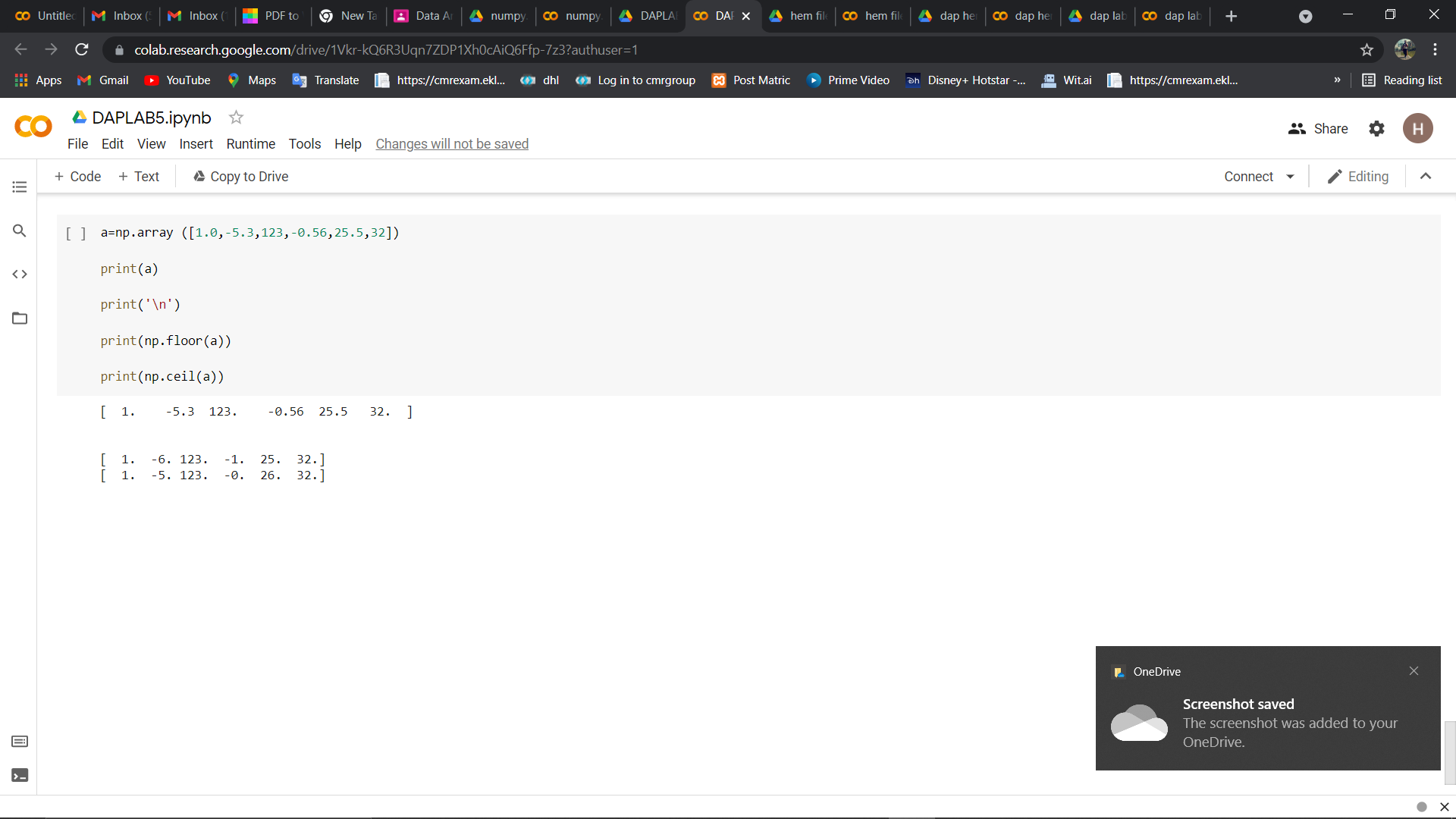
****

****

**NumPy in Python**

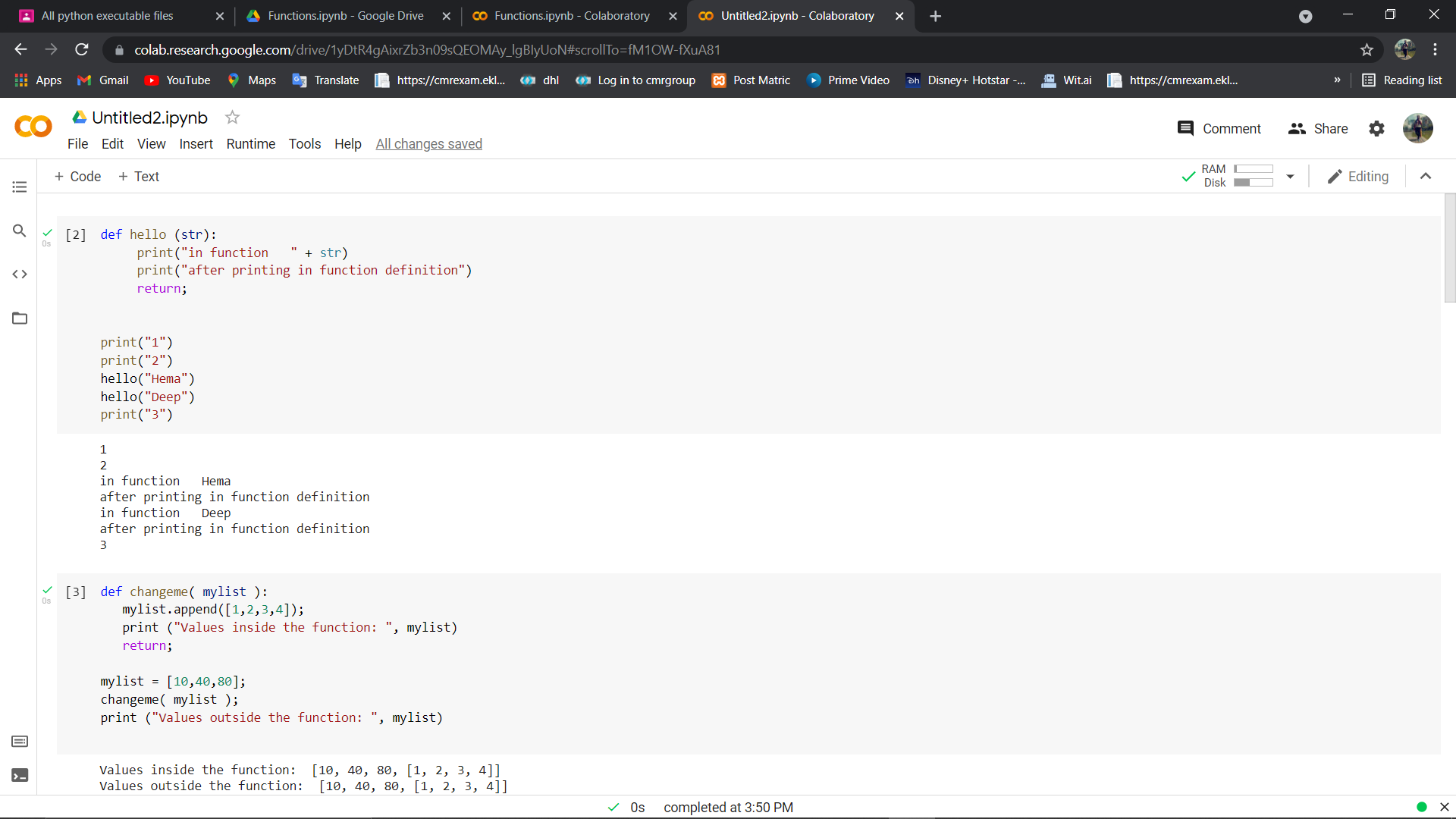
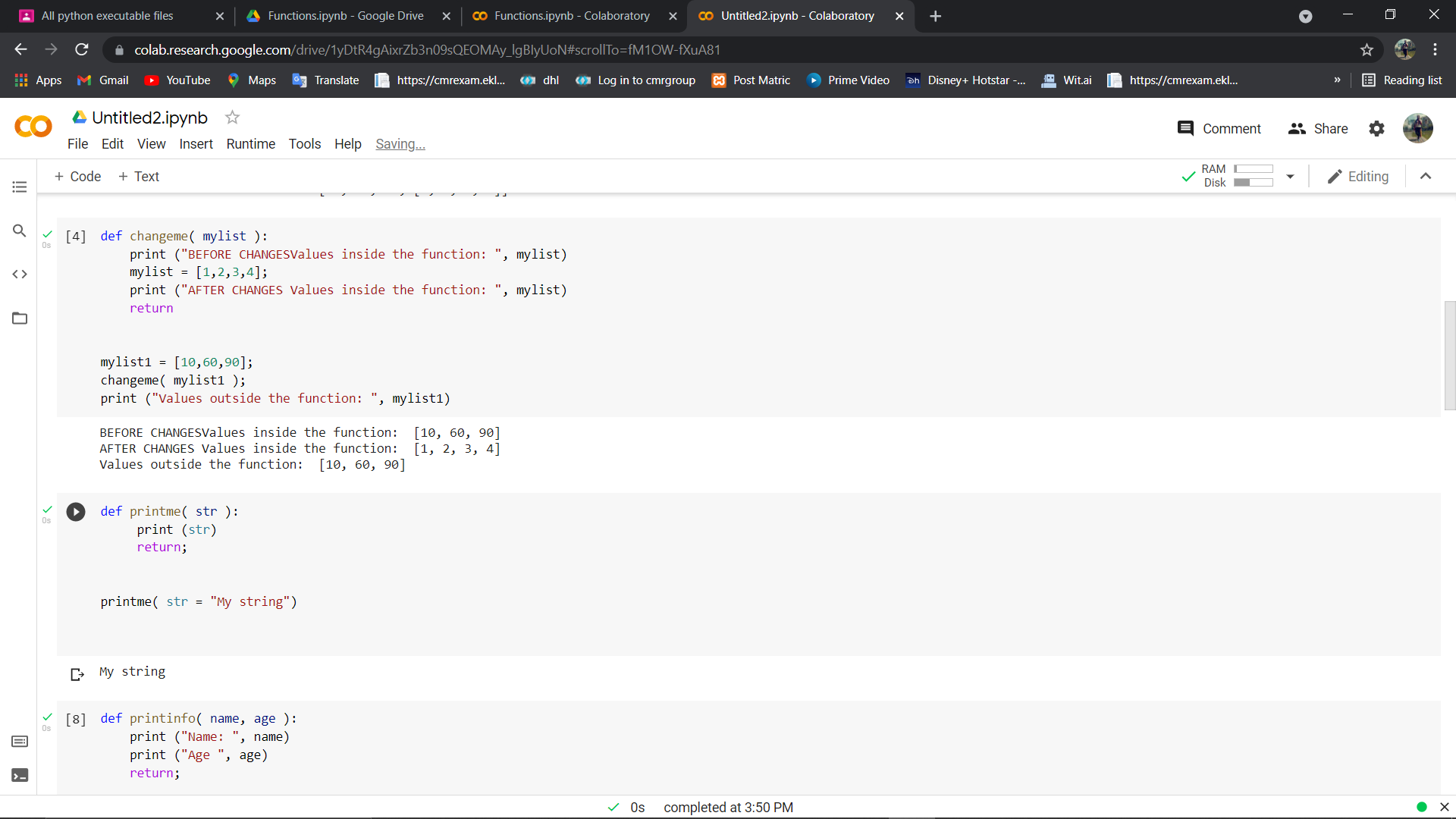
****

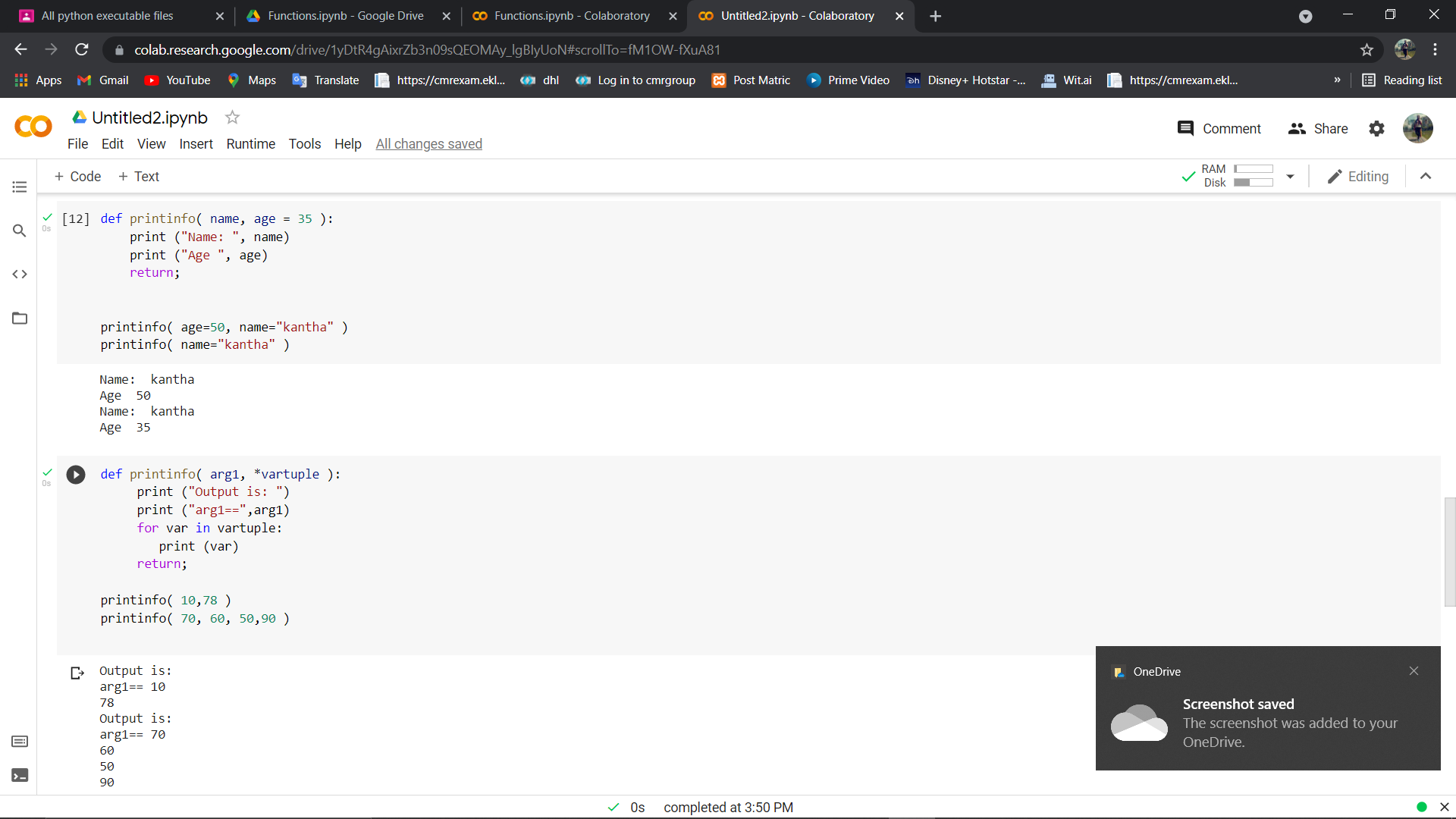
**NumPy Arithmetic OP in Python**

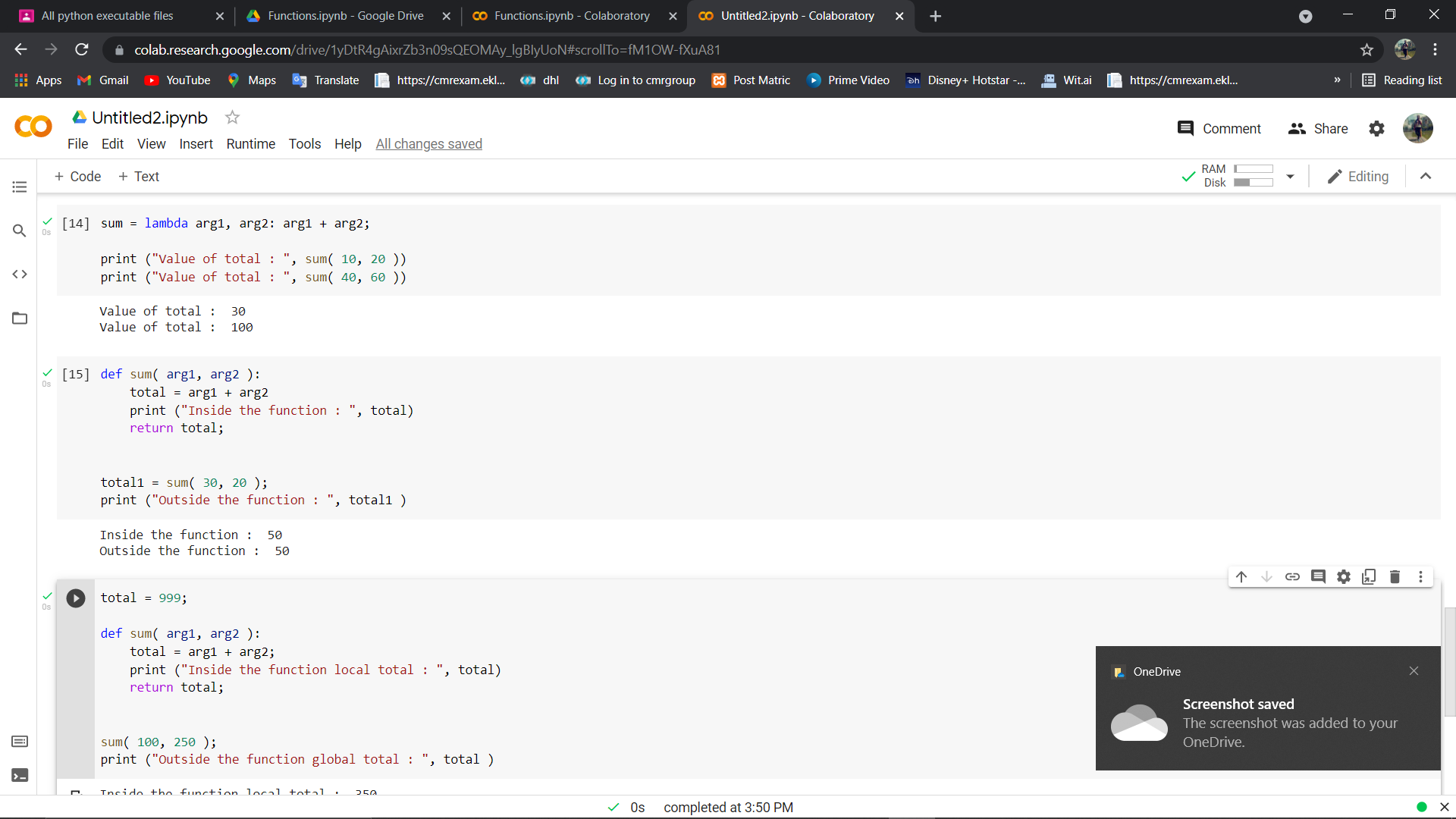
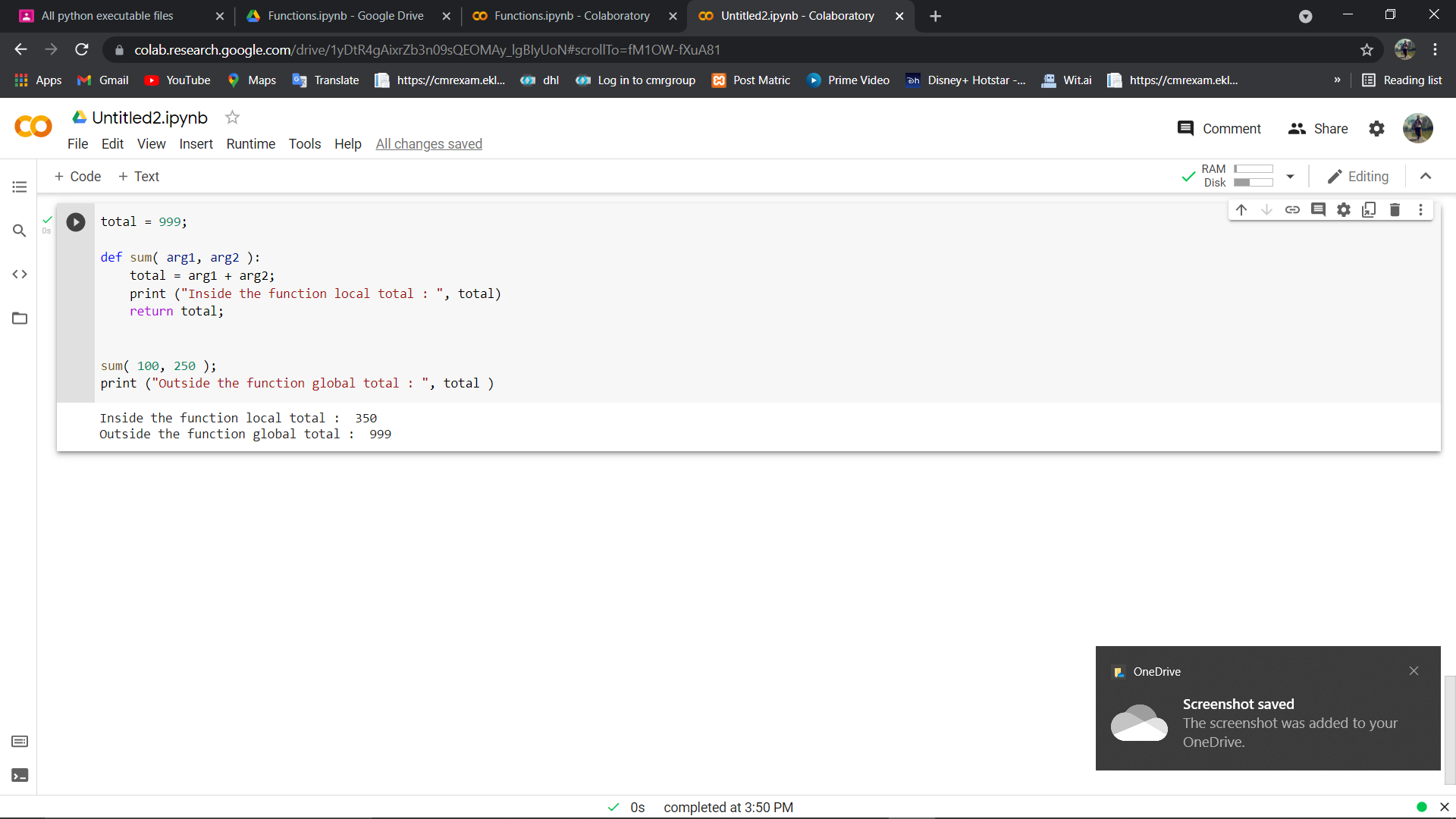
****

**FUNCTIONS**

**Objective:** Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal,… You can find an overview with more of these functions [here](https://docs.python.org/3/library/functions.html).User-Defined Functions (UDFs), which are functions that users create to help them out.Anonymous functions, which are also called lambda functions because they are not declared with the standard def keyword

****

****

****