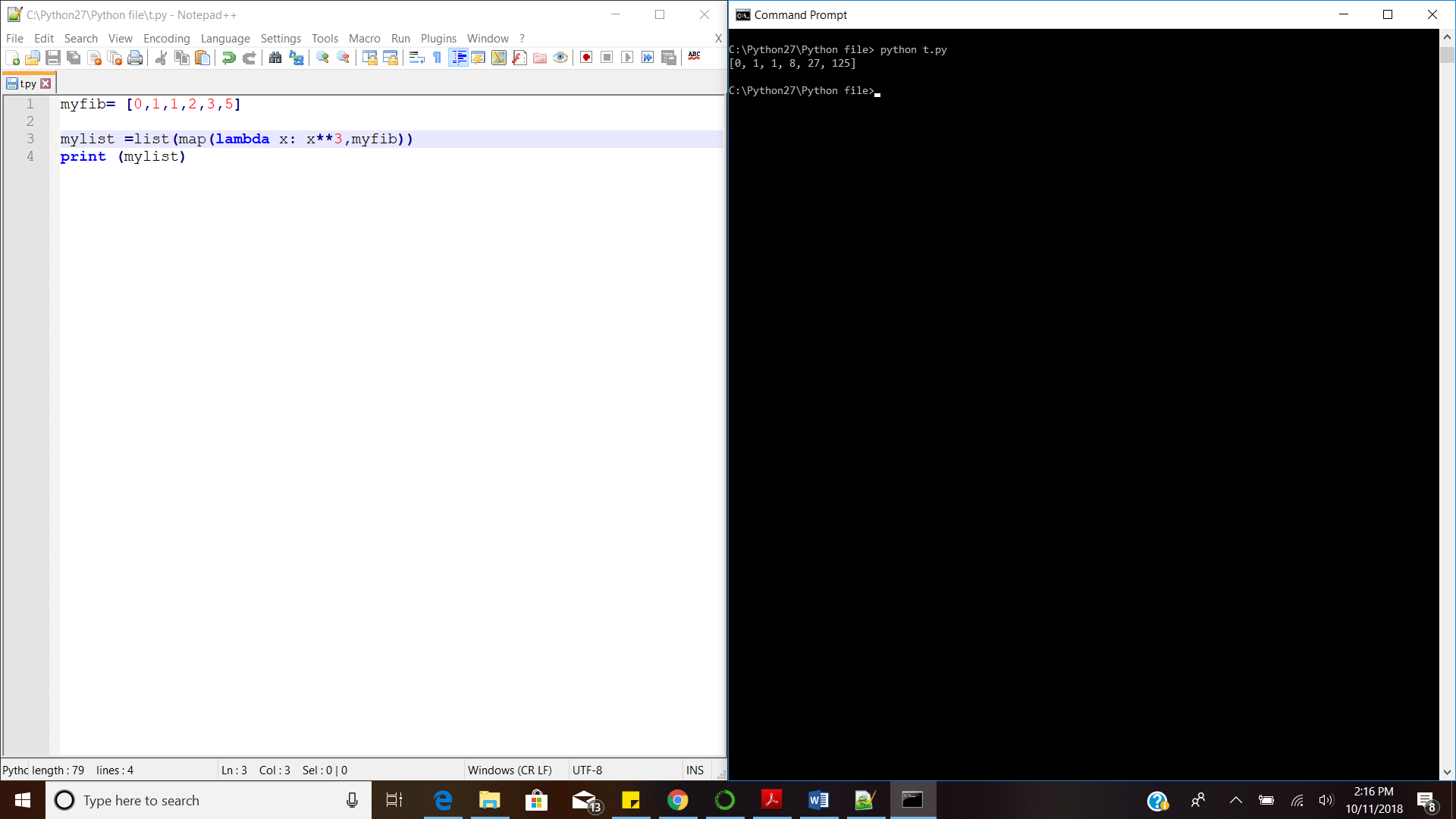
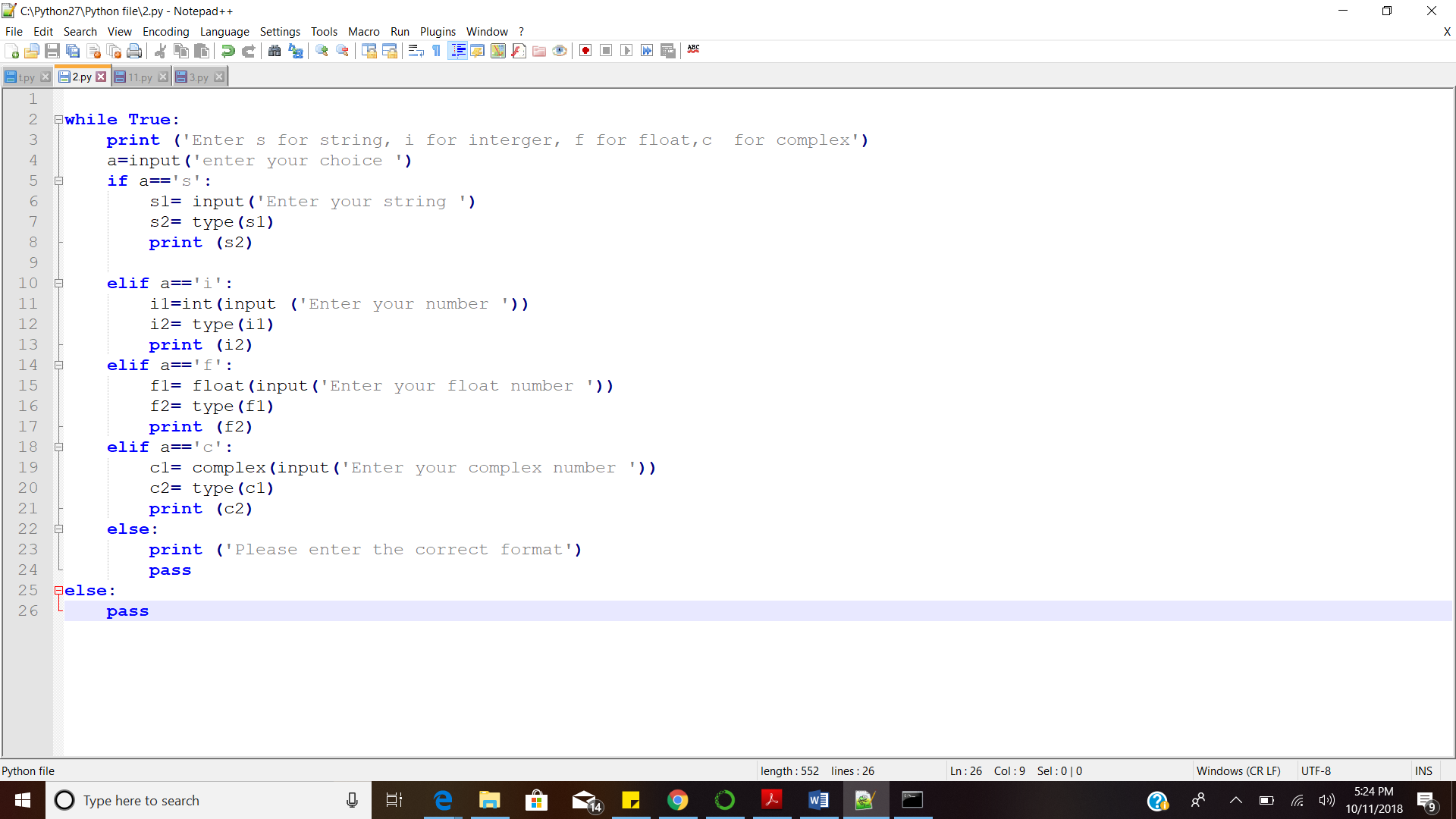
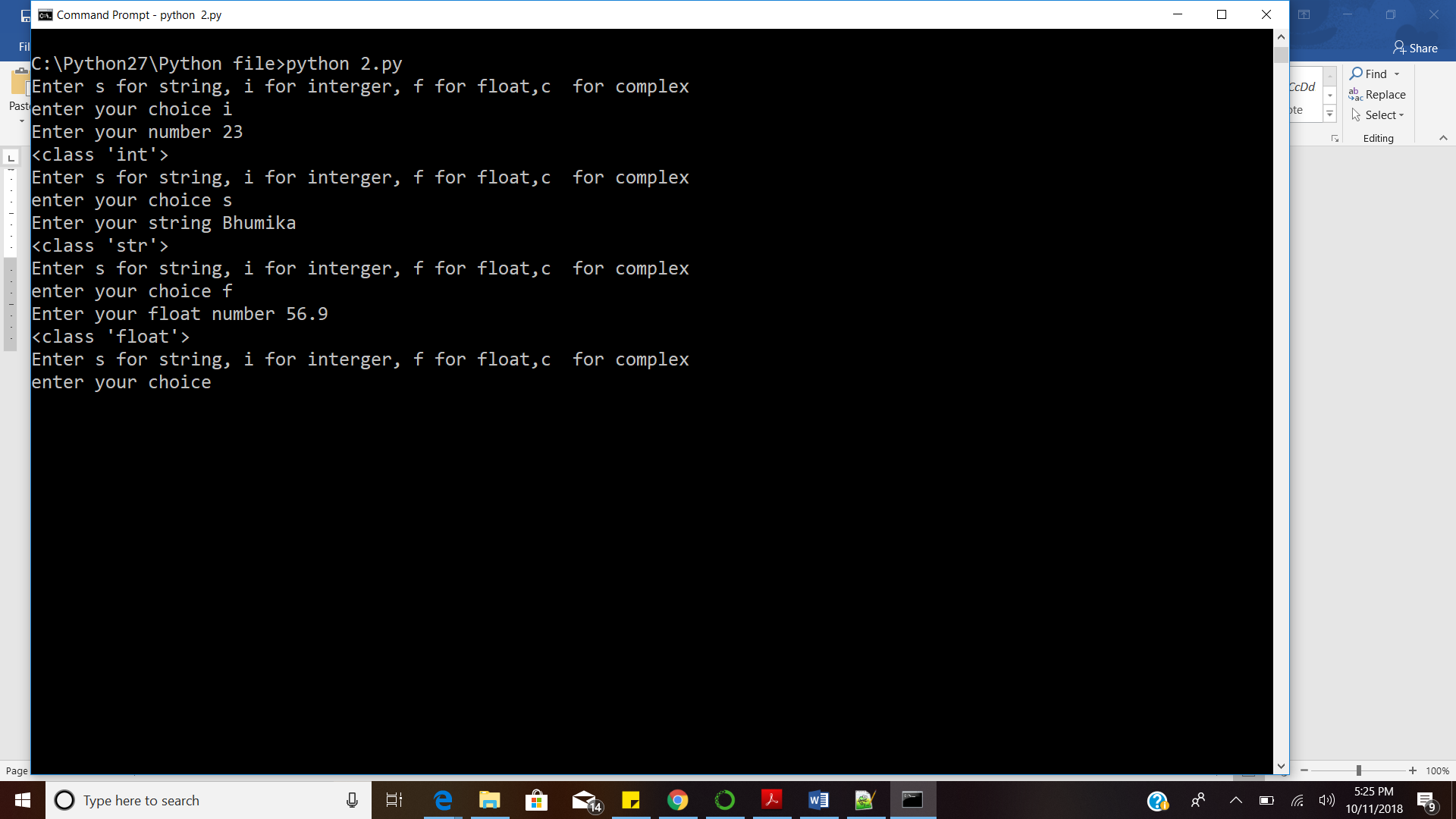
1. **Write a program which should return the list on a single line containing the cubes of the first 6 Fibonacci numbers. HINT: Use Lambda function.**



1. **Write a program to take input from the user in following formats and also check their type by using type function.**

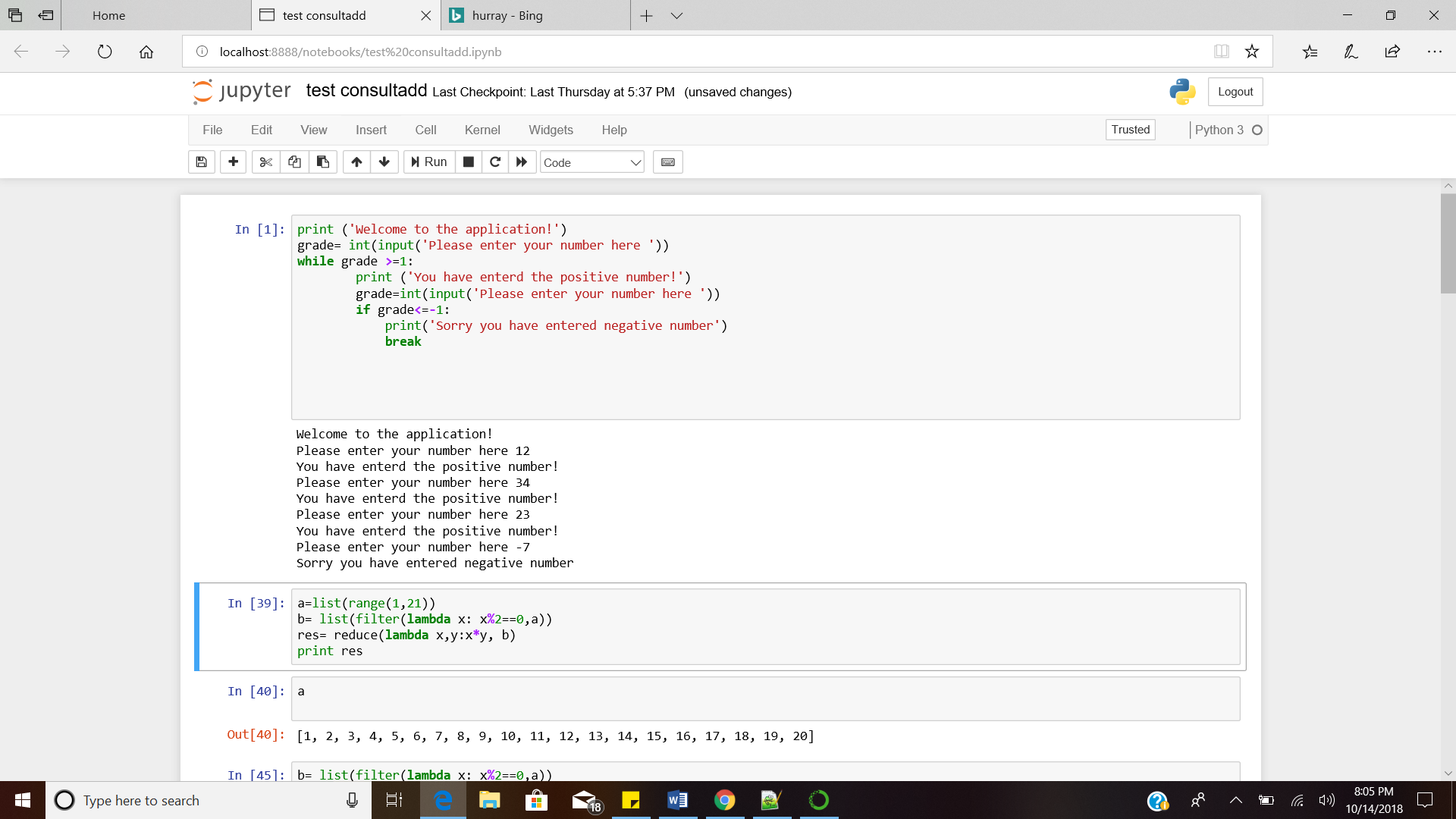
**Formats: Integer, Float, String & Complex**



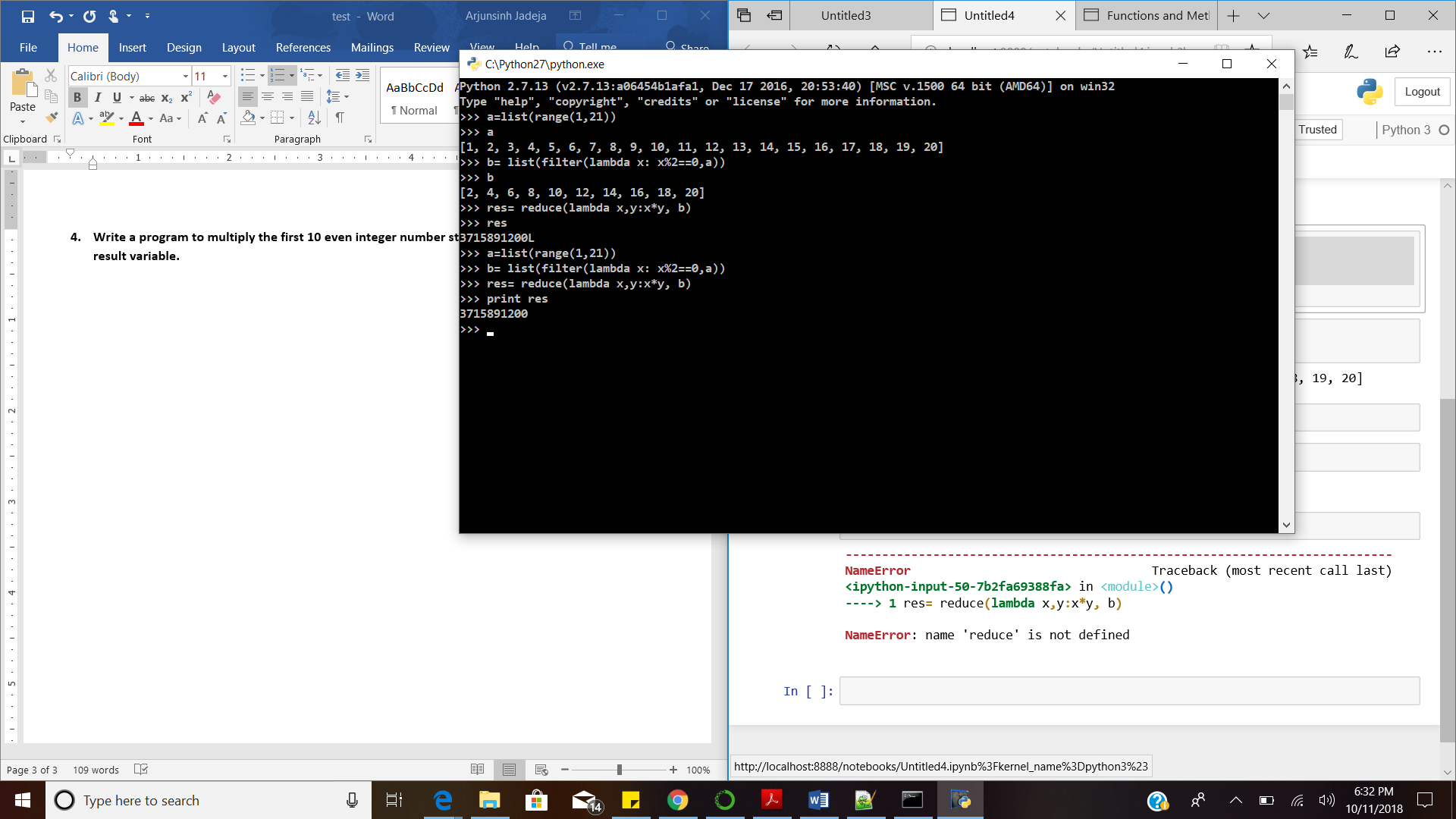


1. **Write a program for the following conditions:**

* **If a user enters a negative number it should break the loop**
* **If a user enters any positive value it will run the loop for infinite.**

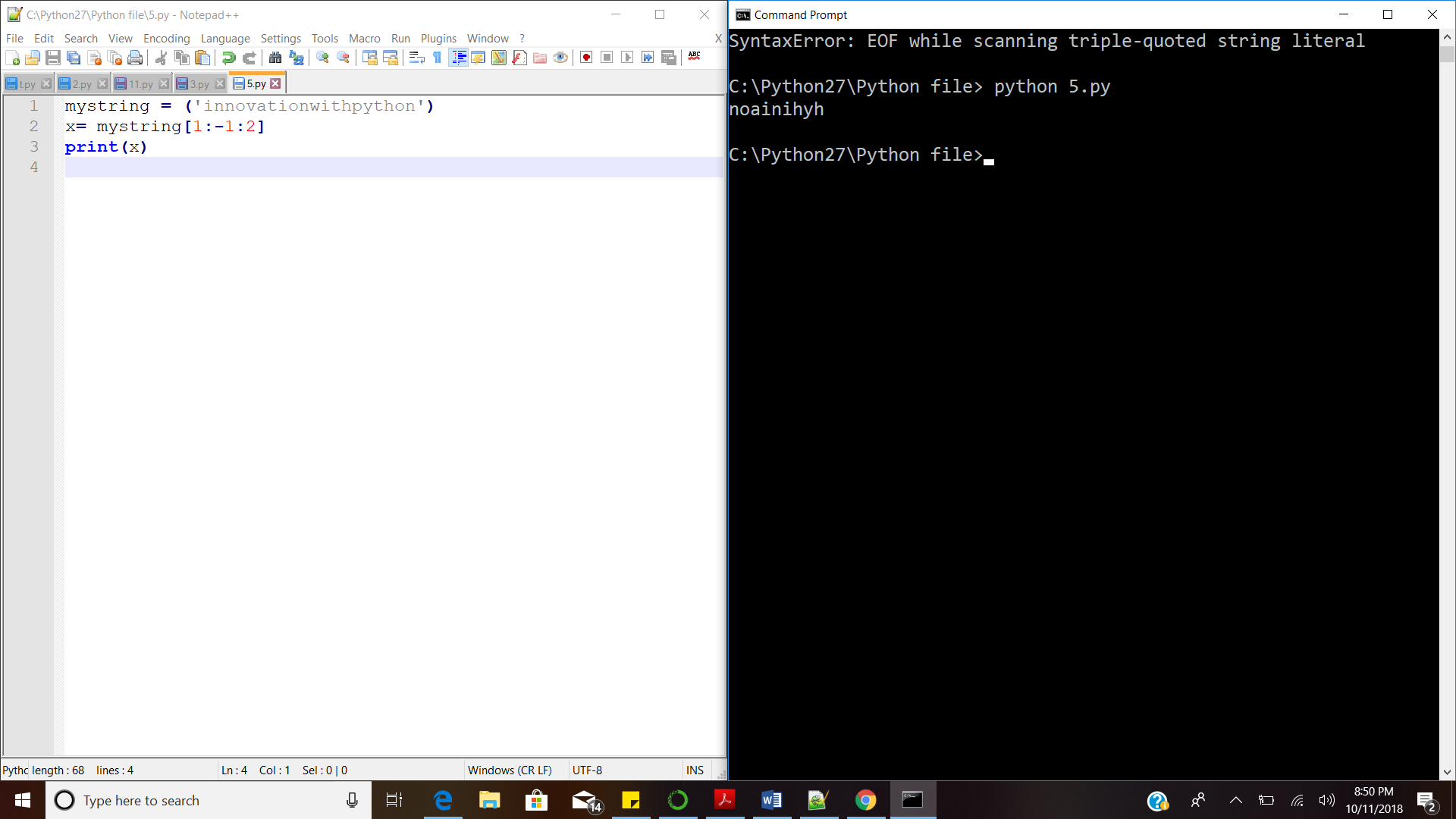


1. **Write a program to multiply the first 10 even integer number starting from 1 & store them in result variable.**



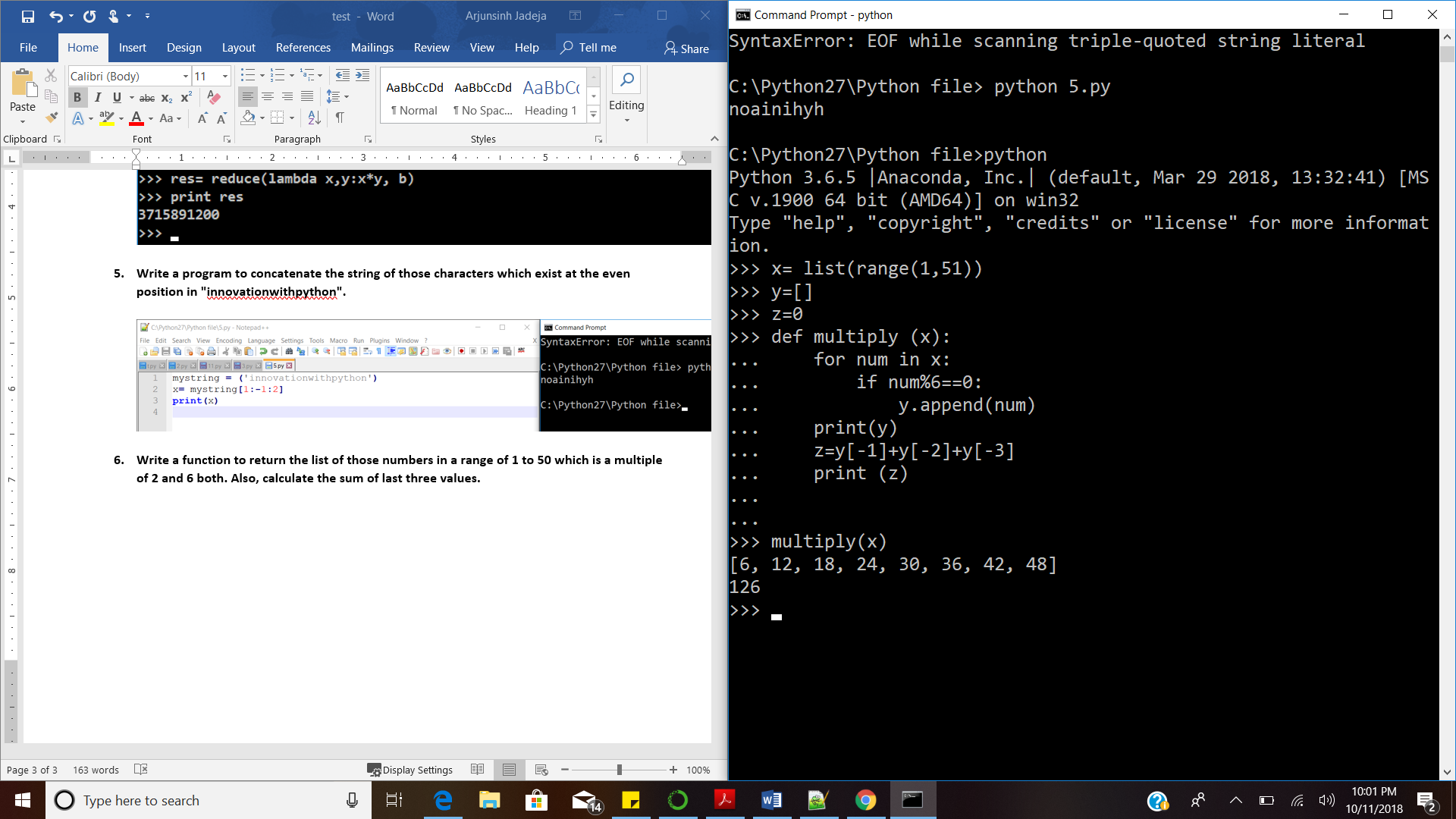
1. **Write a program to concatenate the string of those characters which exist at the even**

**position in "innovationwithpython".**



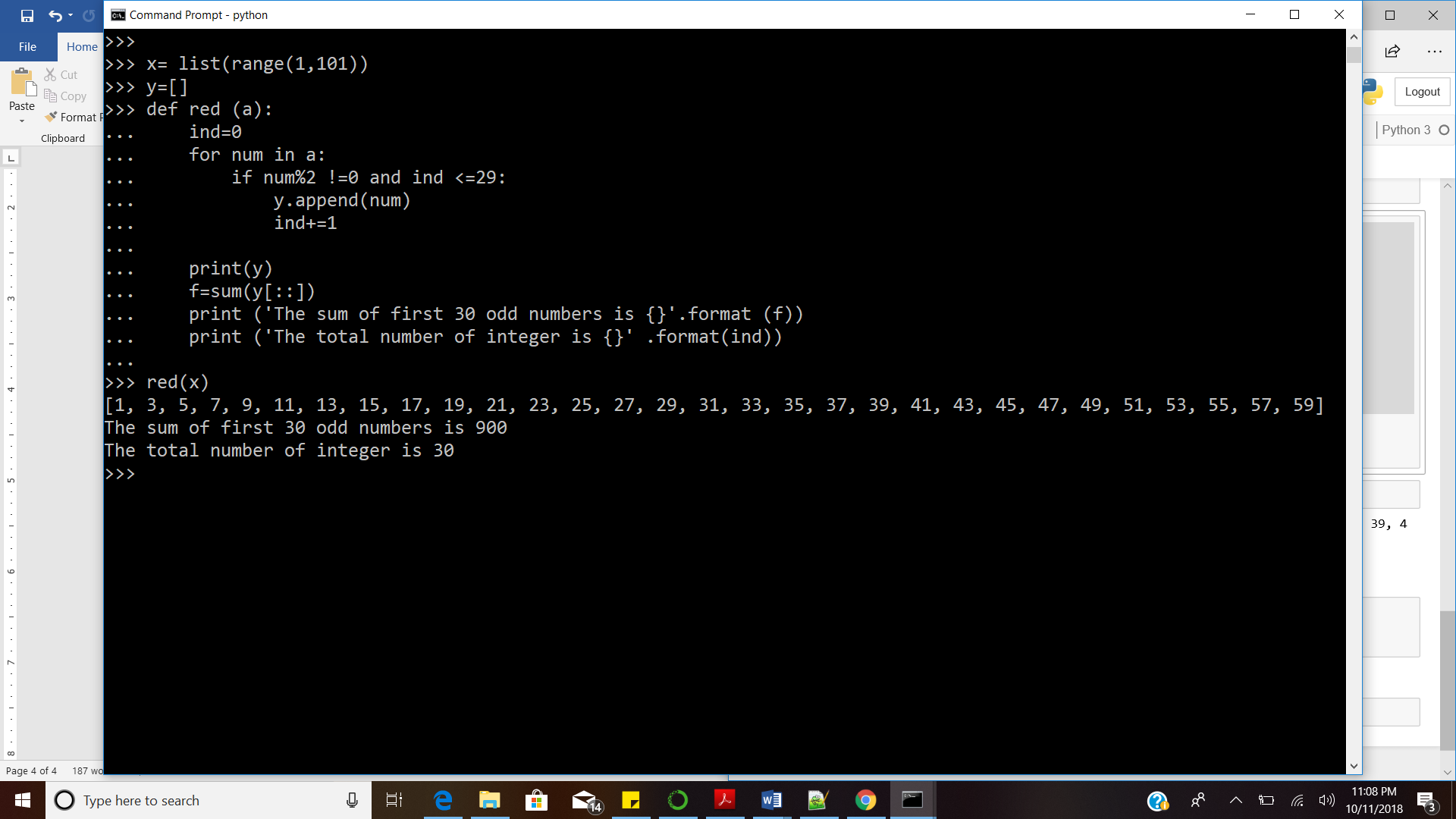
1. **Write a function to return the list of those numbers in a range of 1 to 50 which is a multiple**

**of 2 and 6 both. Also, calculate the sum of last three values.**



1. **Write the higher order function reduce to calculate the total sum of first 30 odd values in he**

**range of 1 to 100.**



1. **What will be the output of the following:**

**new\_list=[ 1 , 2 , 3 , 4 , 5 , 6 , [ "Riyaz" , "Ul" , "Haque" , 7 ] , 8 , 9 , 10 ]**

**--- new\_list [ -4 ]**

['Riyaz', 'Ul', 'Haque', 7]

**--- new\_list [ 4 ]**

5

**--- new\_list [ 6 ] [ 1 ]**

“Ul”

**--- new\_list . append ( [ "new" ] )**

[1, 2, 3, 4, 5, 6, ['Riyaz', 'Ul', 'Haque', 7], 8, 9, 10, ['new']]

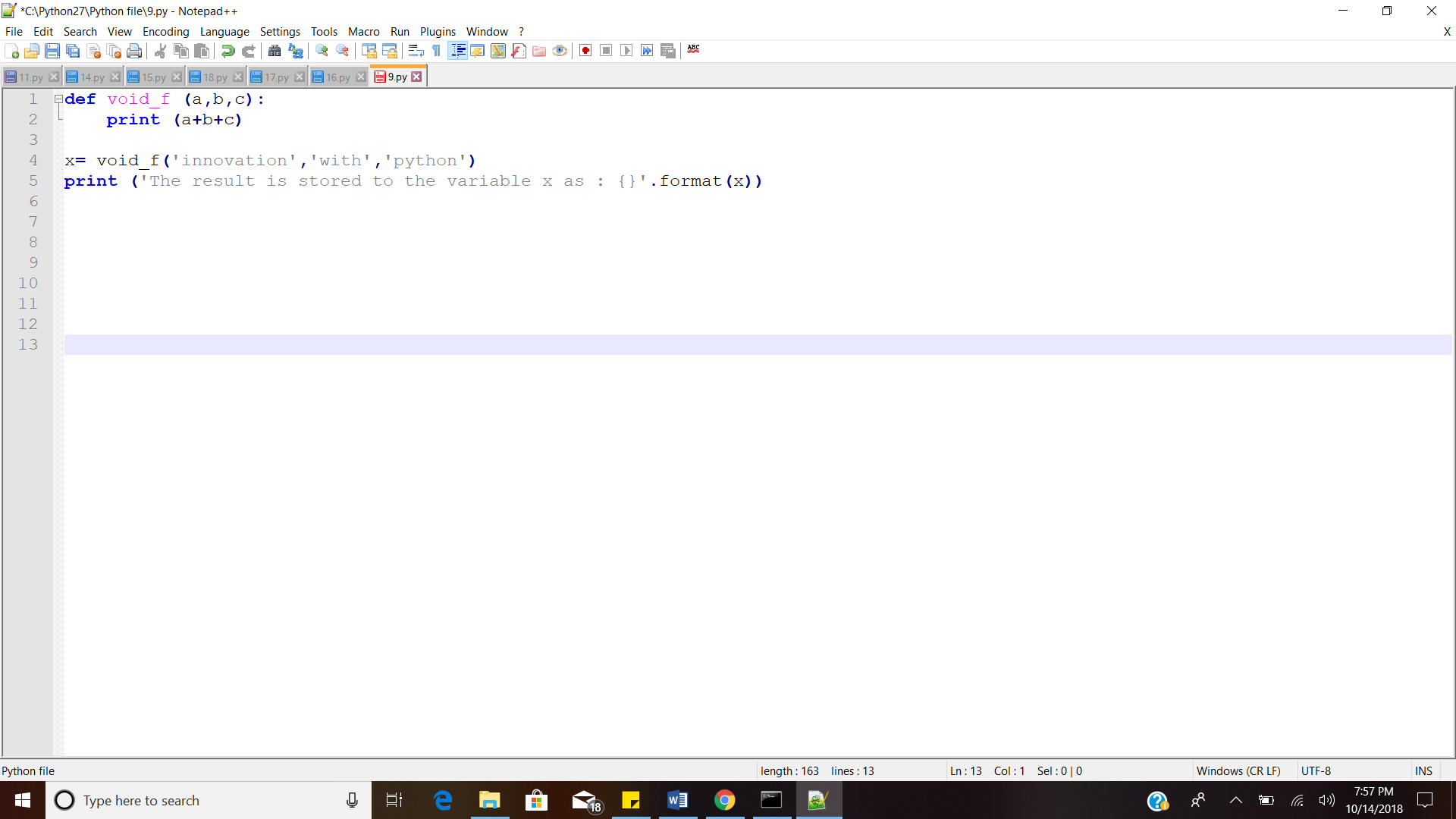
**--- new\_list . sort( )**

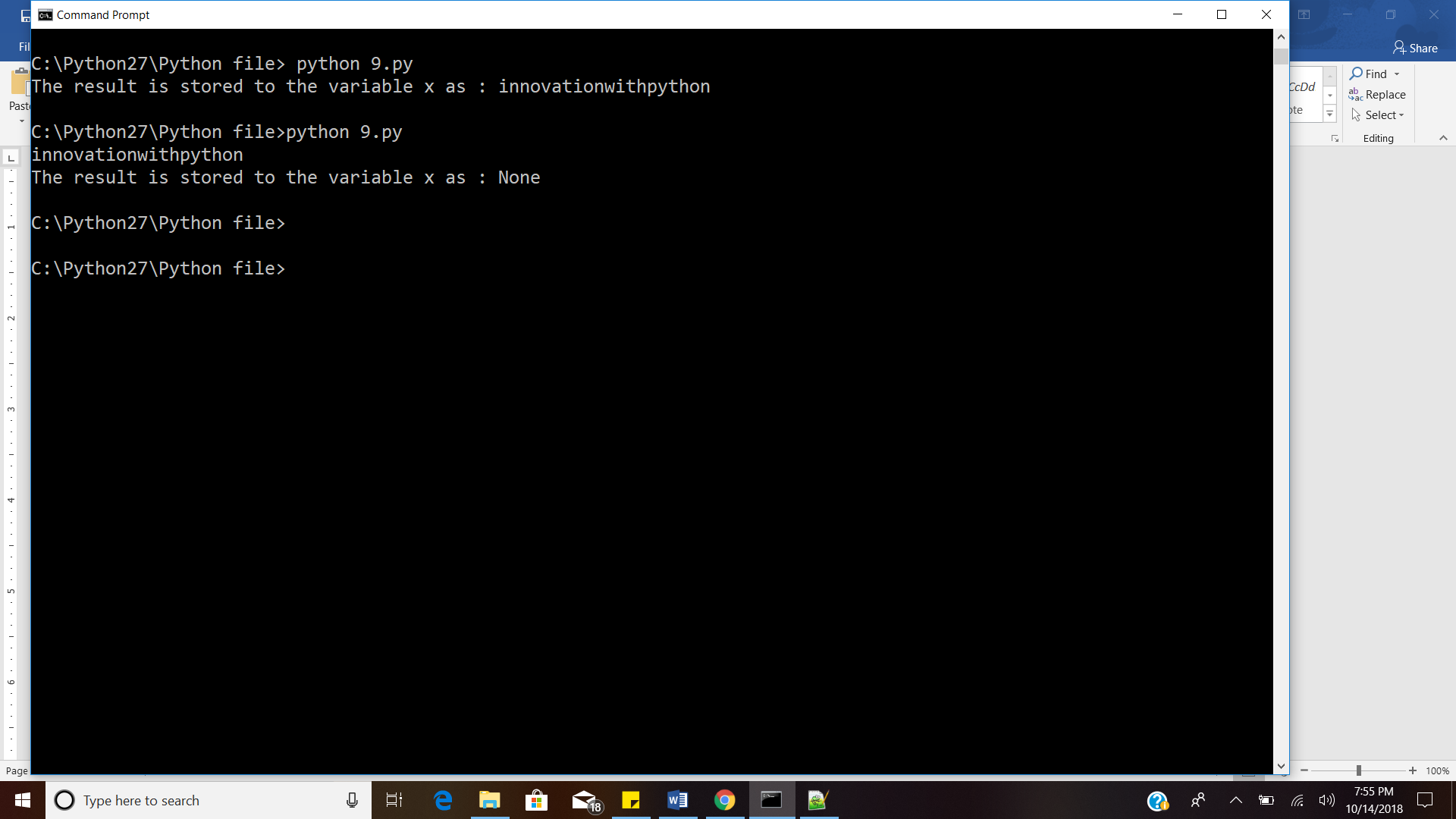
[1, 2, 3, 4, 5, 6, 8, 9, 10, ['Riyaz', 'Ul', 'Haque', 7], ['new']]

1. **Write a function that takes three arguments which after concatenation should print**

**"innovationwithpython" . This is a void type function.**

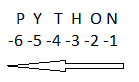
In order to create void type function, **print** statement was used instead of the return statement.





1. **What is**
2. **Negative indexing in python:** Negative indexing is giving the negative index position. The order of the negative indexing is reverse, it starts from the end. The indexing start with -1.

e.g. Negative indexing for the string ‘PYTHON’



1. **Packing and unpacking:** Packing packs all the arguments into single variable. Packing will form tuple, which is immutable. The tuple can be converted into list to modify.

Unpacking: All elements of the sequence can be passed as different parameter.

1. **Mutable and immutable:** Mutable is the characteristic of data type/data structure which allows them to change. The examples of mutable data structure are: list, dictionaries.

Immutable does not allow the data type or data structure to change. The example of immutable data type is string and immutable data structure is tuple ().

1. **Append and Extend:** Both append and extend are used to add values to list.

Append increase the length of list by +1.

Extend increase the length of the sequence by the number of elements present in the group added.

e.g. x=[1,2,3,4,5]

x.append[a,b] ----- x=[1,2,3,4,5,[a,b]]-------increase length by 1

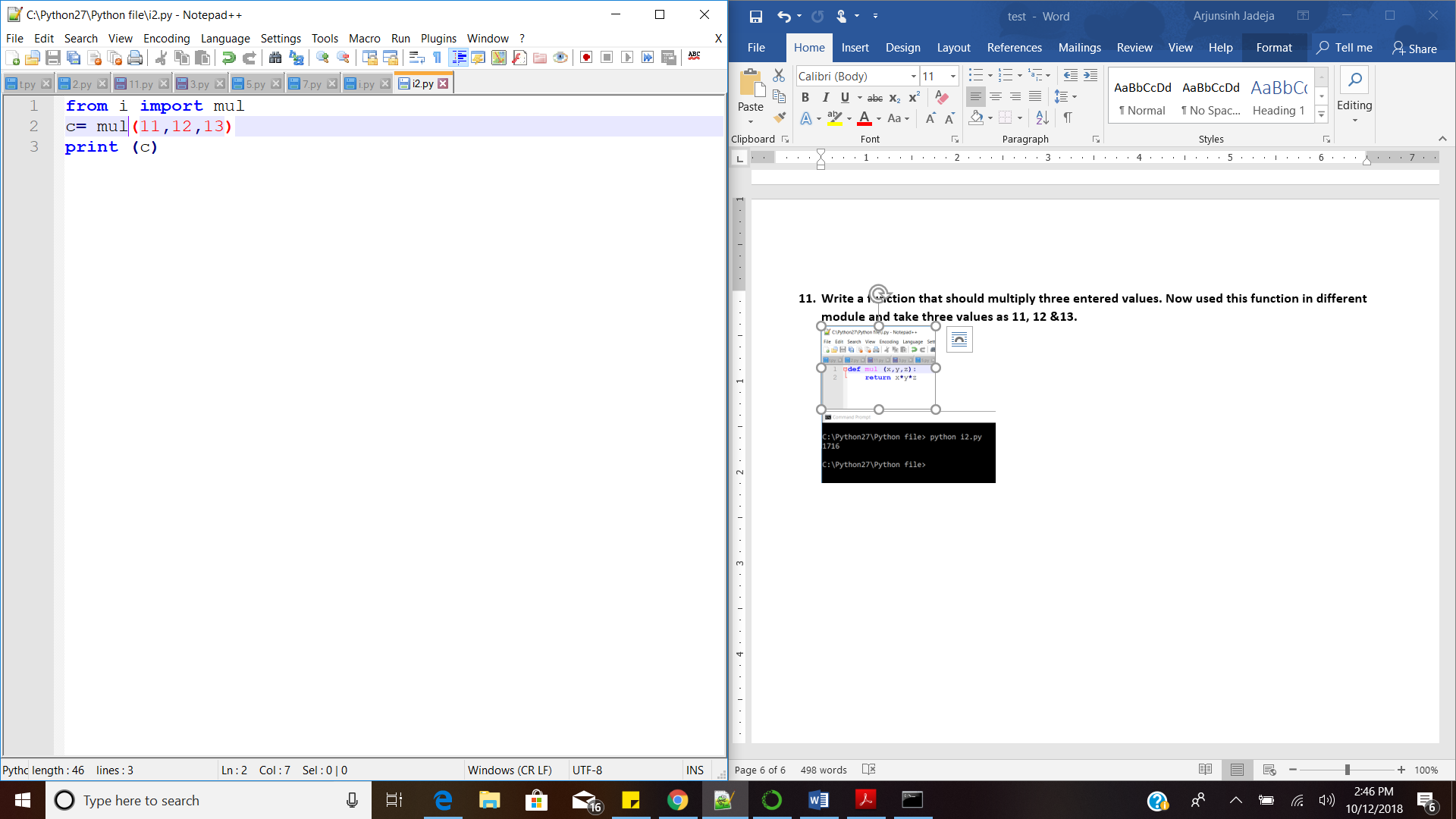
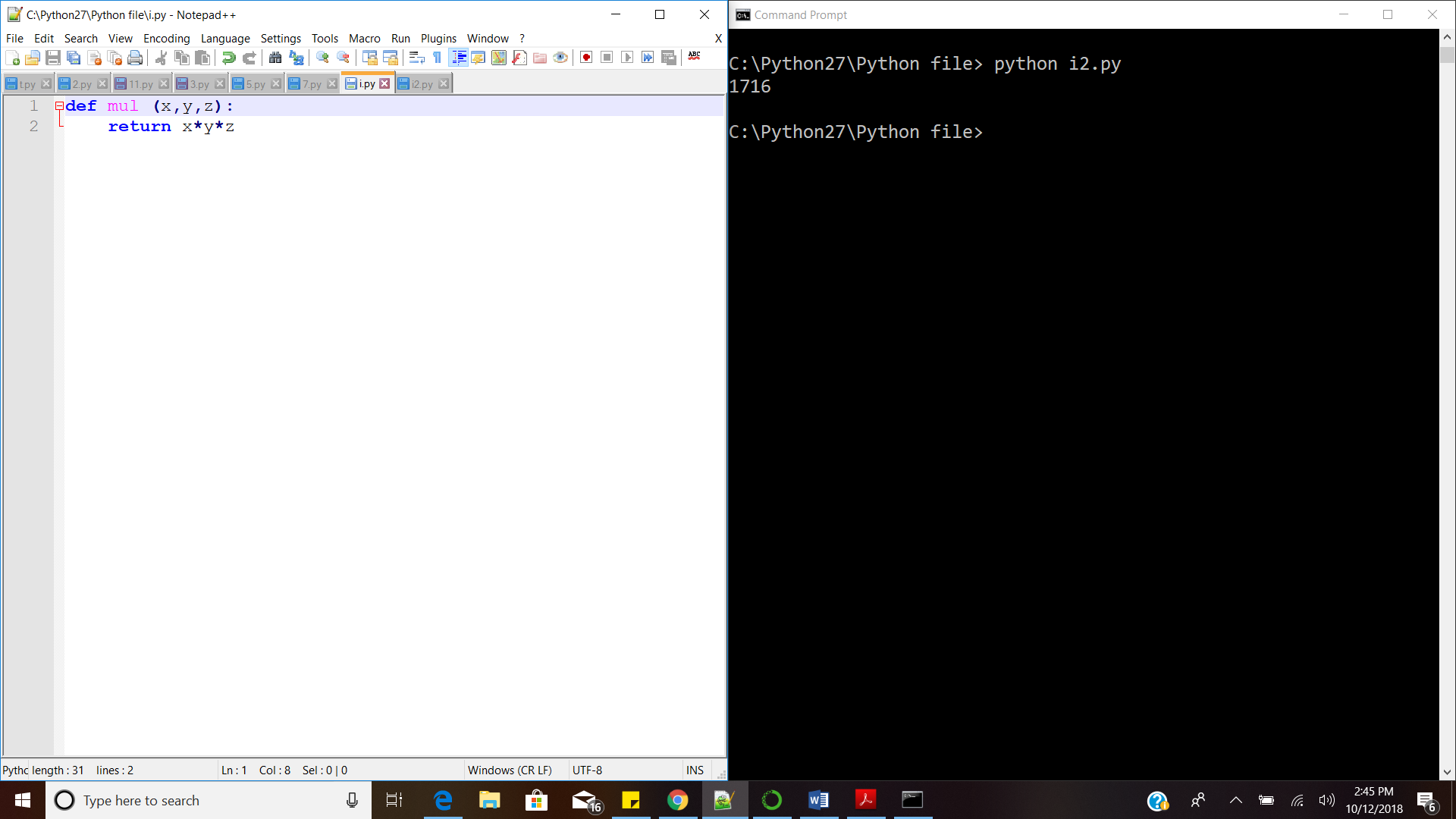
x.extend[a,b]------- x=[1,2,3,4,5,a,b]---------increase length by 2

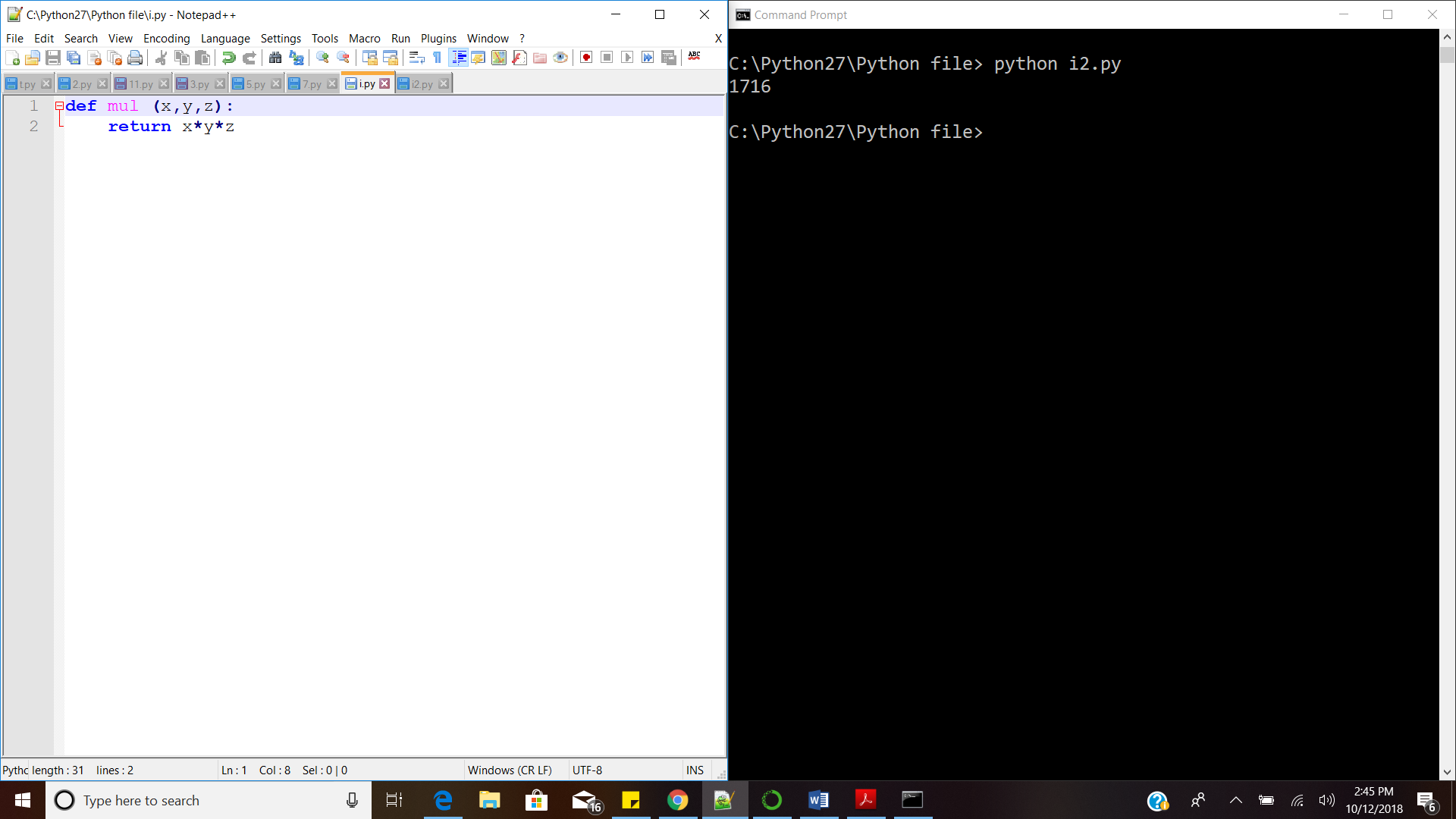
1. **Pickling and Unpickling**

Pickling: The process of converting the python object into the byte stream in order to store it in a file or database, transport data over the network or maintain program state across session.

Unpickling: The process of converting the byte stream to object structure.

1. **Write a function that should multiply three entered values. Now used this function in different module and take three values as 11, 12 &13.**





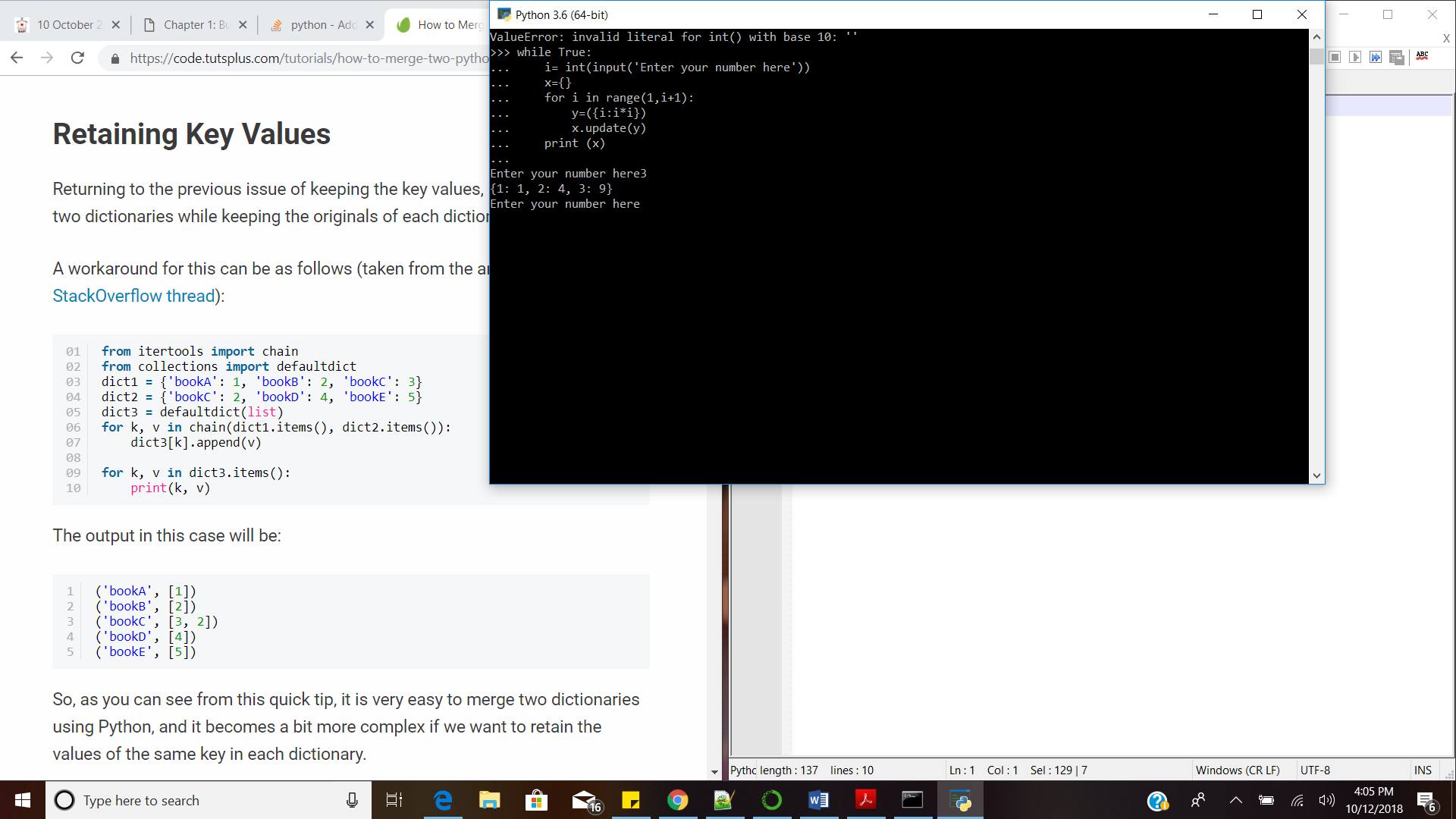
1. **Write a program to generate a dictionary that contains (i, i\*i) such that is integral number between 1 and n (both included). and then the program should print the dictionary.**

**Suppose the following input is supplied to the program:**

**4**

**Then, the output should be:**

**{1: 1, 2: 4, 3: 9, 4: 16}**



1. **What is split function in python?**

* Split function gives output in the form of list of string. For the list x, x.split() syntax is used to get the output.
* The split function separates the words in the string as the elements of the list at specified separator. If no specified separator, it will separate at the spaces.
* e.g.

>>> x= ('Hello this is Python')

>>> y= x.split()

>>> y

['Hello', 'this', 'is', 'Python']

1. **Write a program which accepts a sequence of comma-separated numbers from the console and generate a list and a tuple which contains every number.**

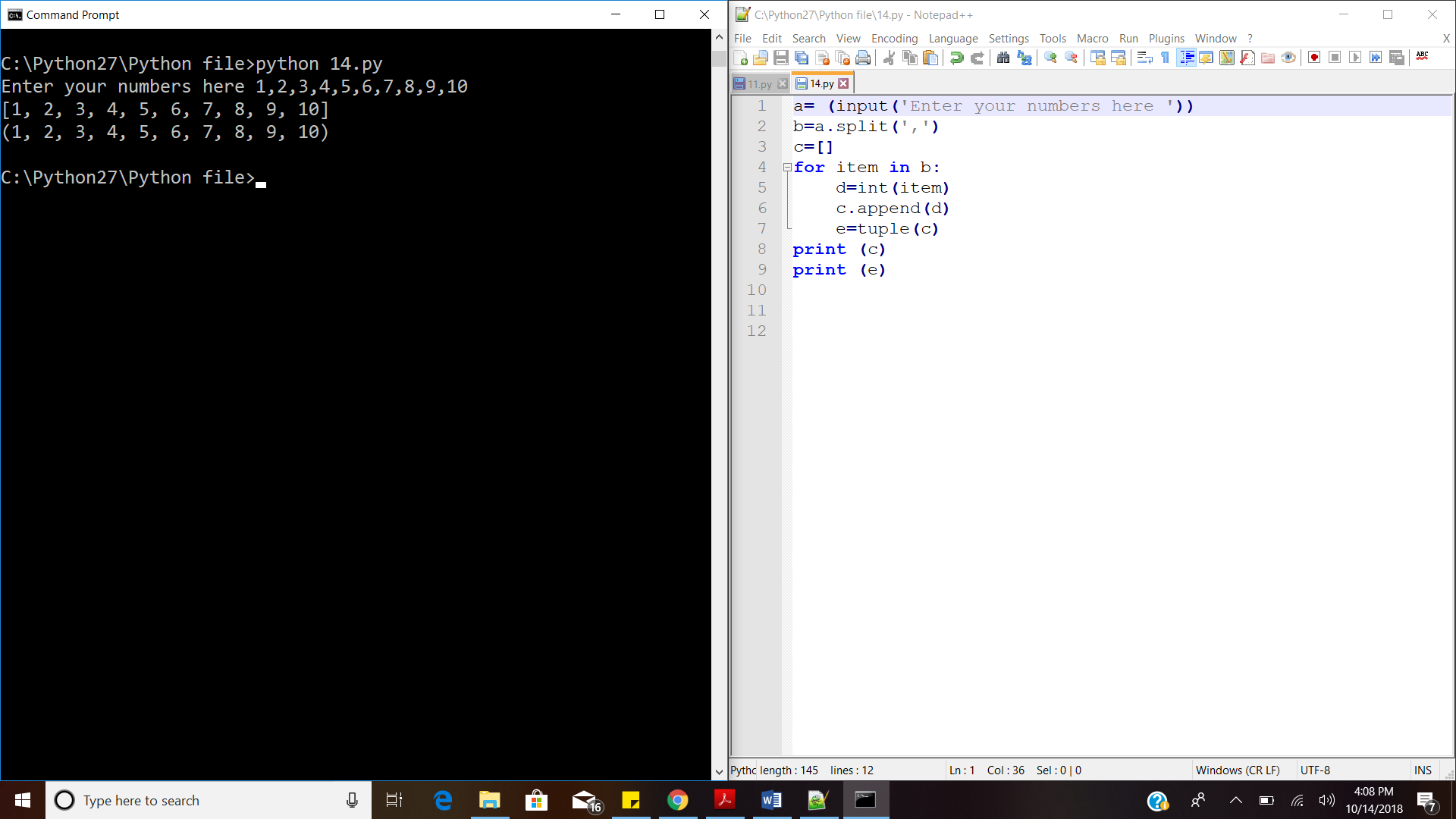
**Suppose the following input is supplied to the program:**

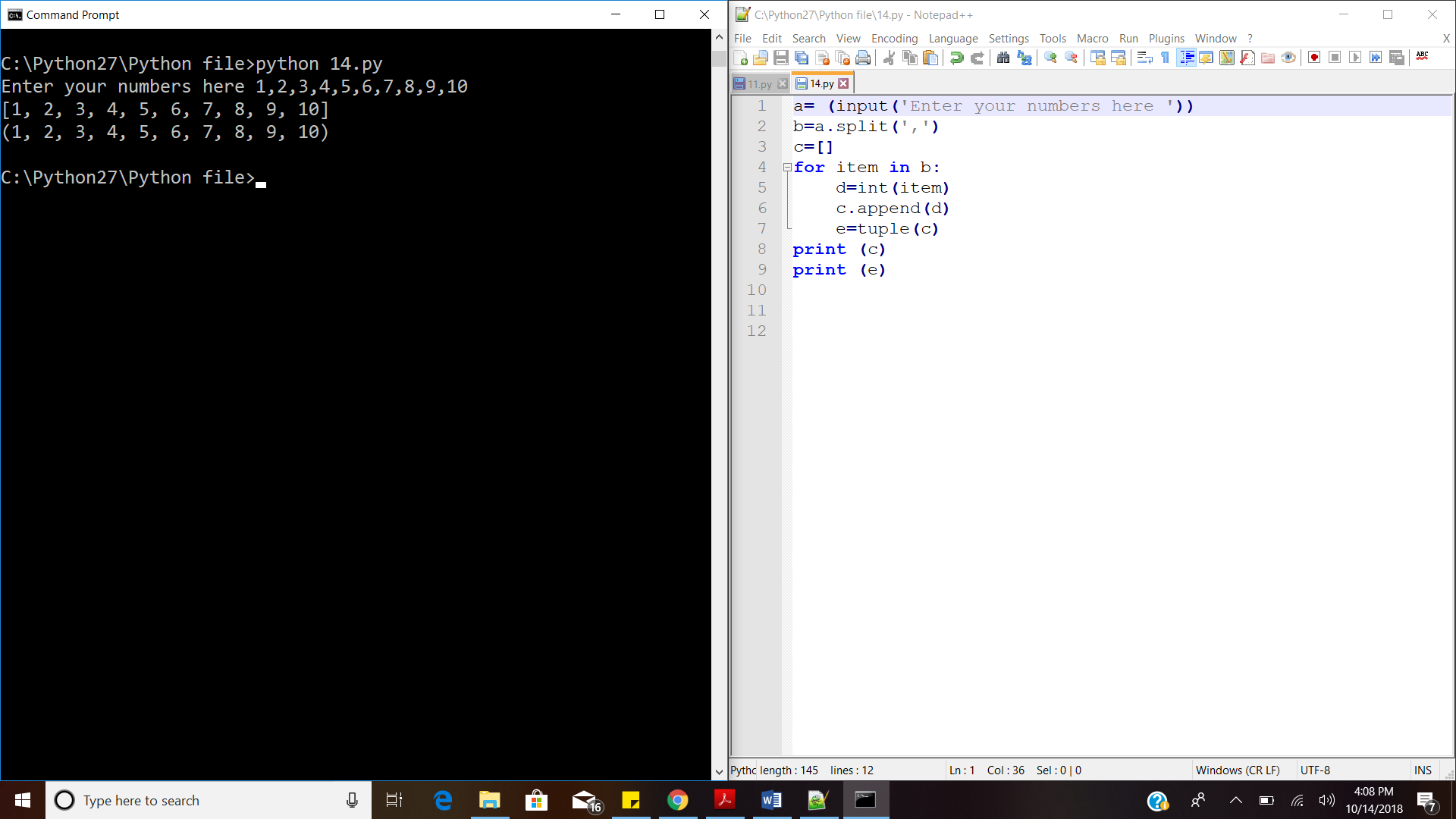
**34,67,55,33,12,98**

**Then, the output should be:**

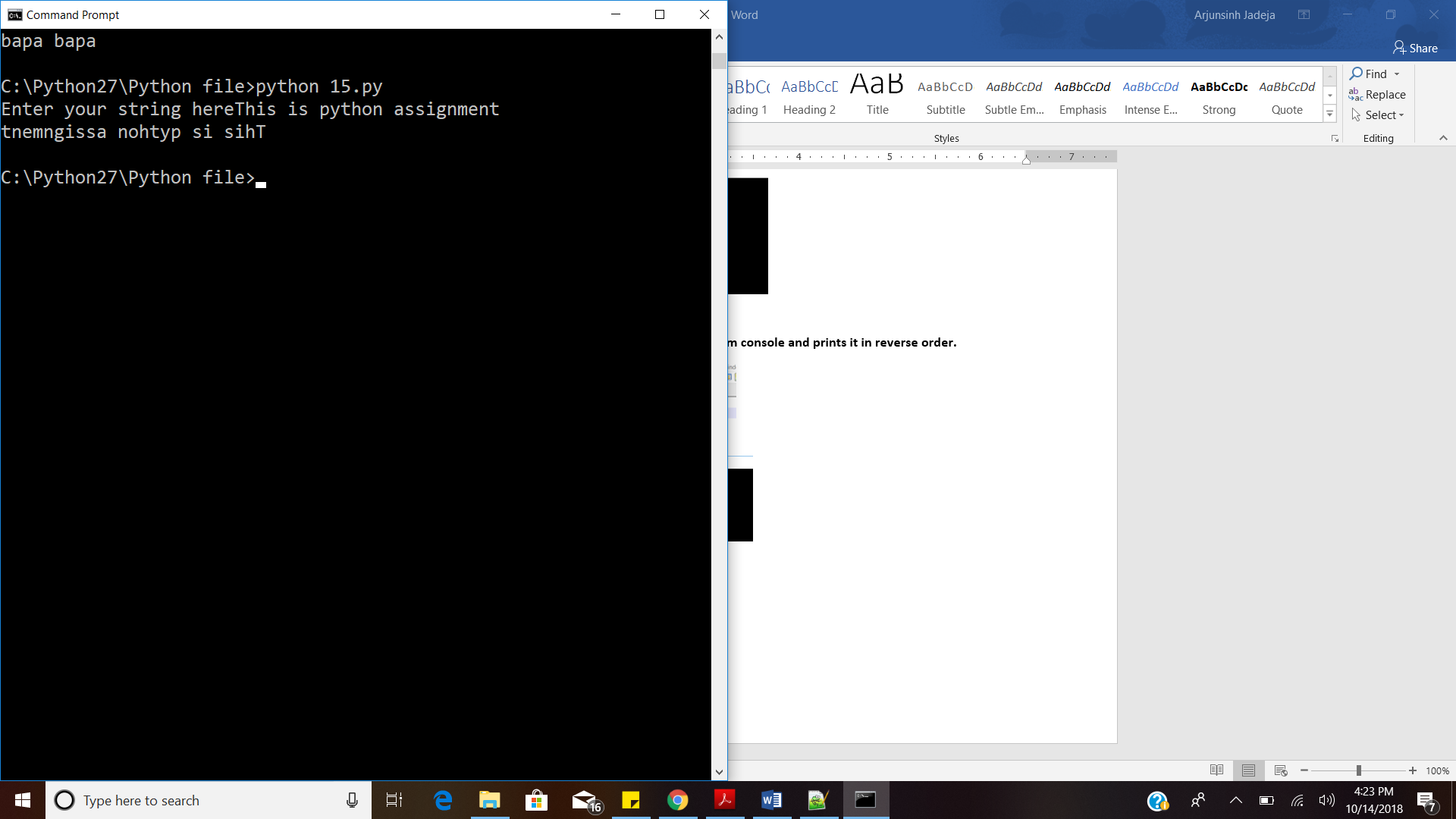
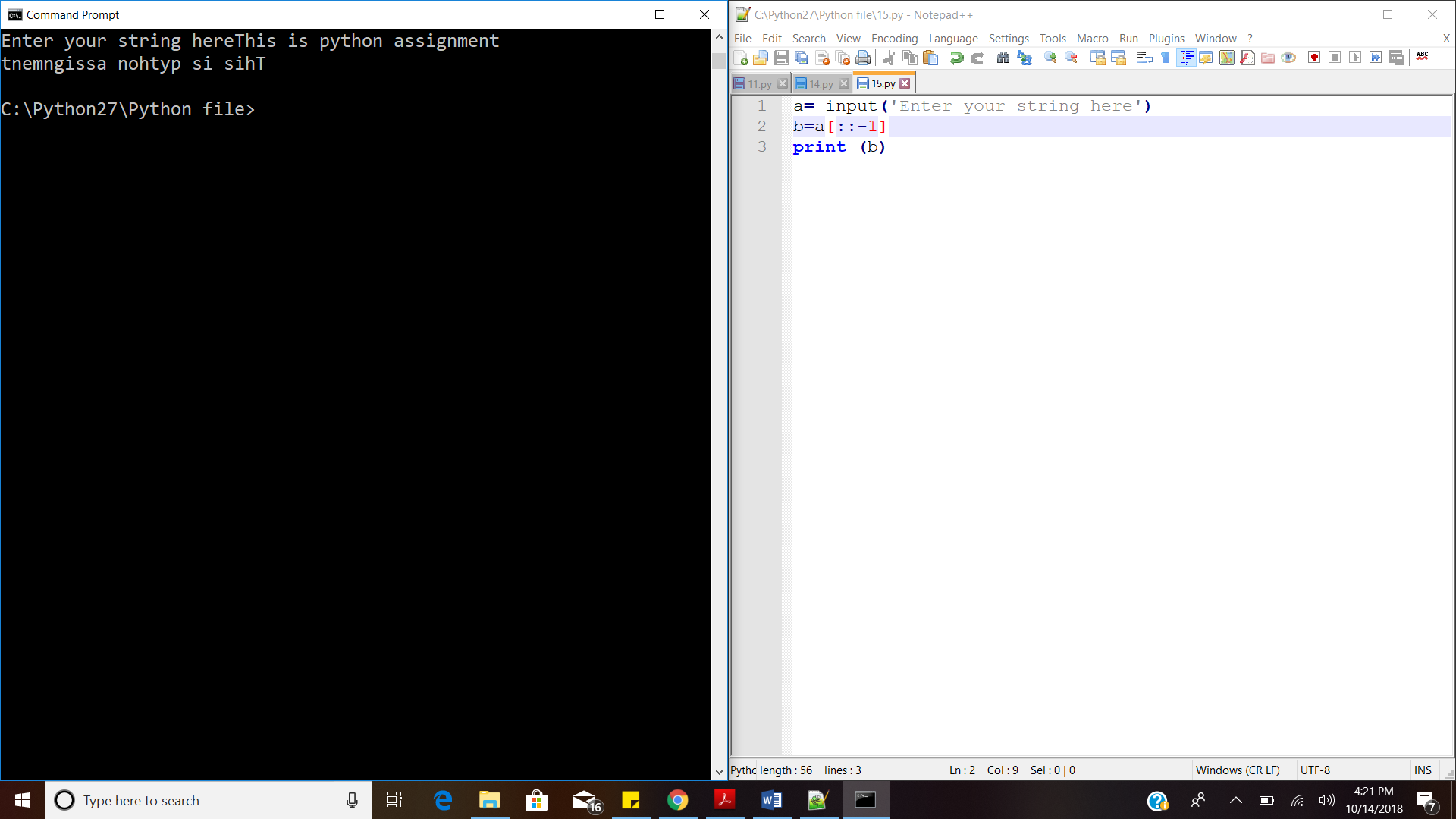
**['34', '67', '55', '33', '12', '98']**

**('34', '67', '55', '33', '12', '98')**

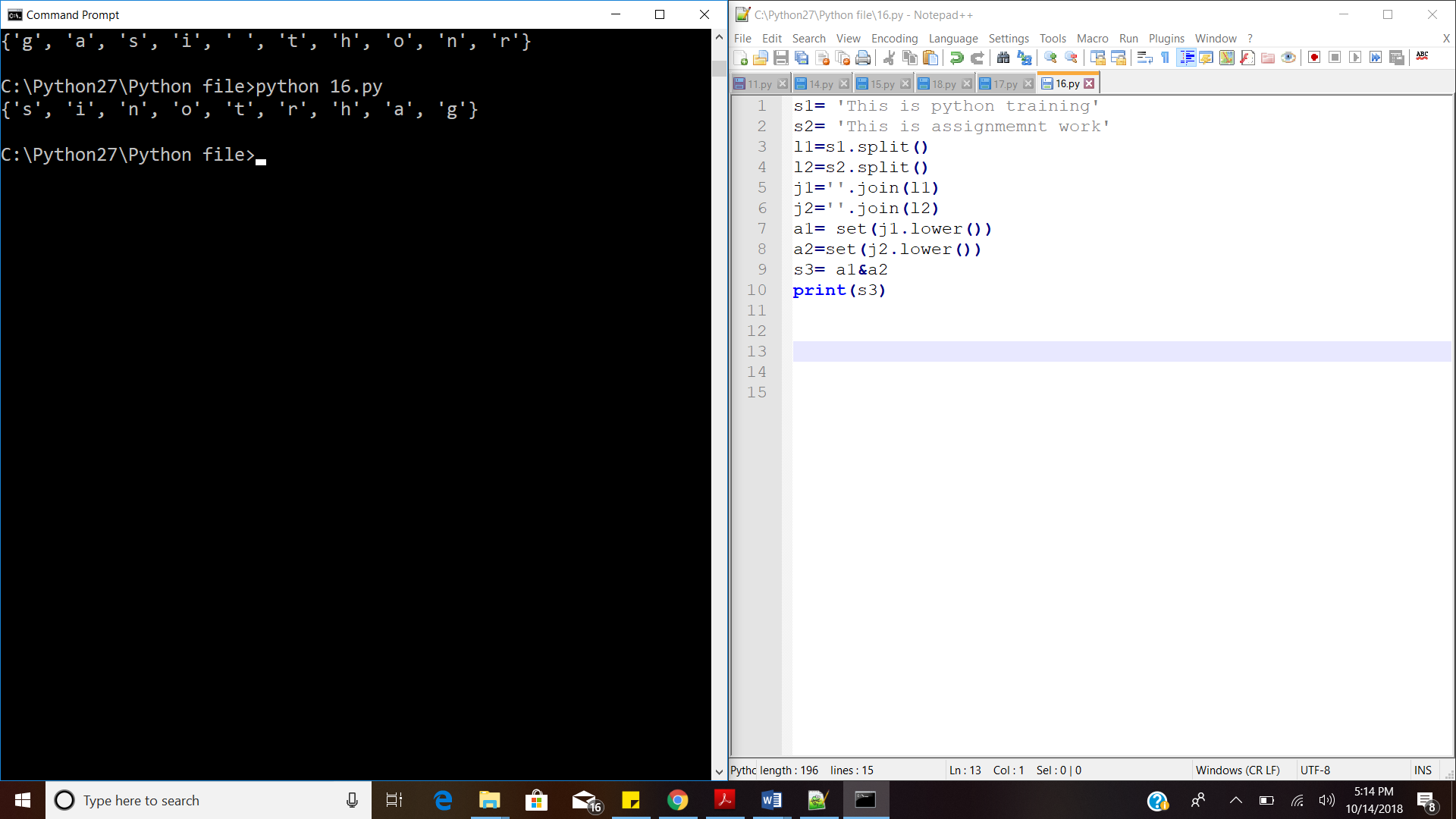


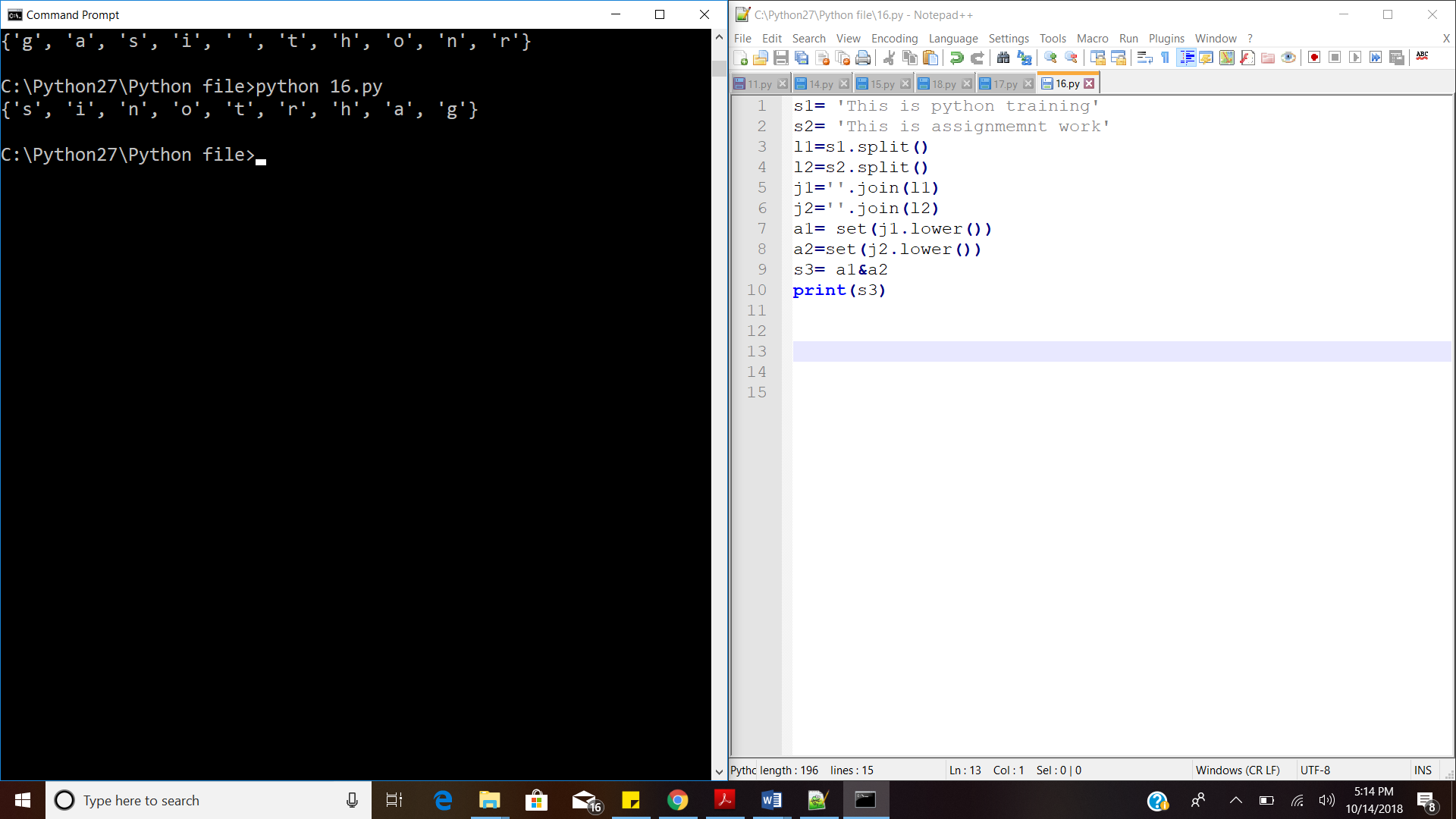


1. **Write a program which accepts a string from console and prints it in reverse order.**

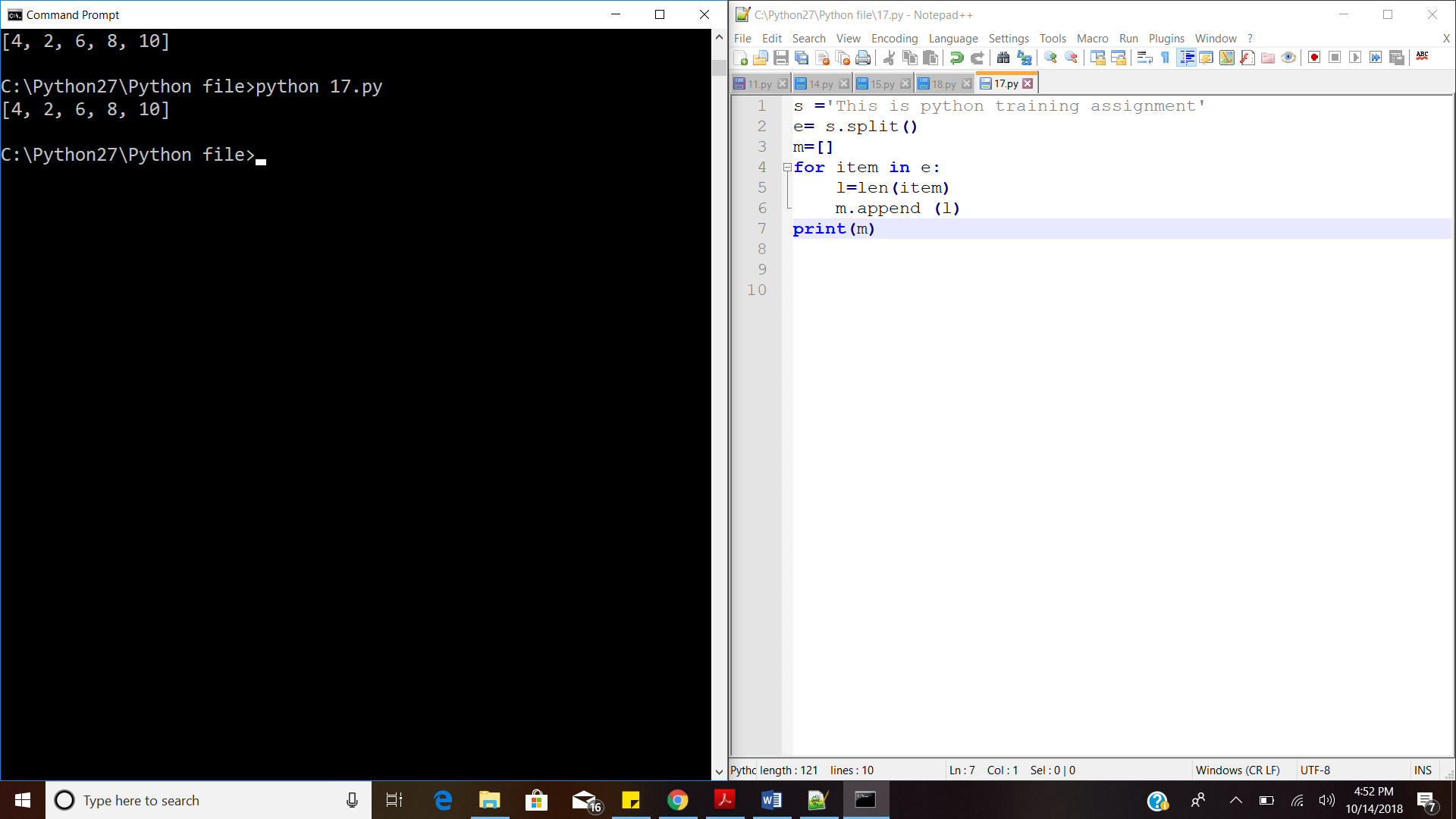


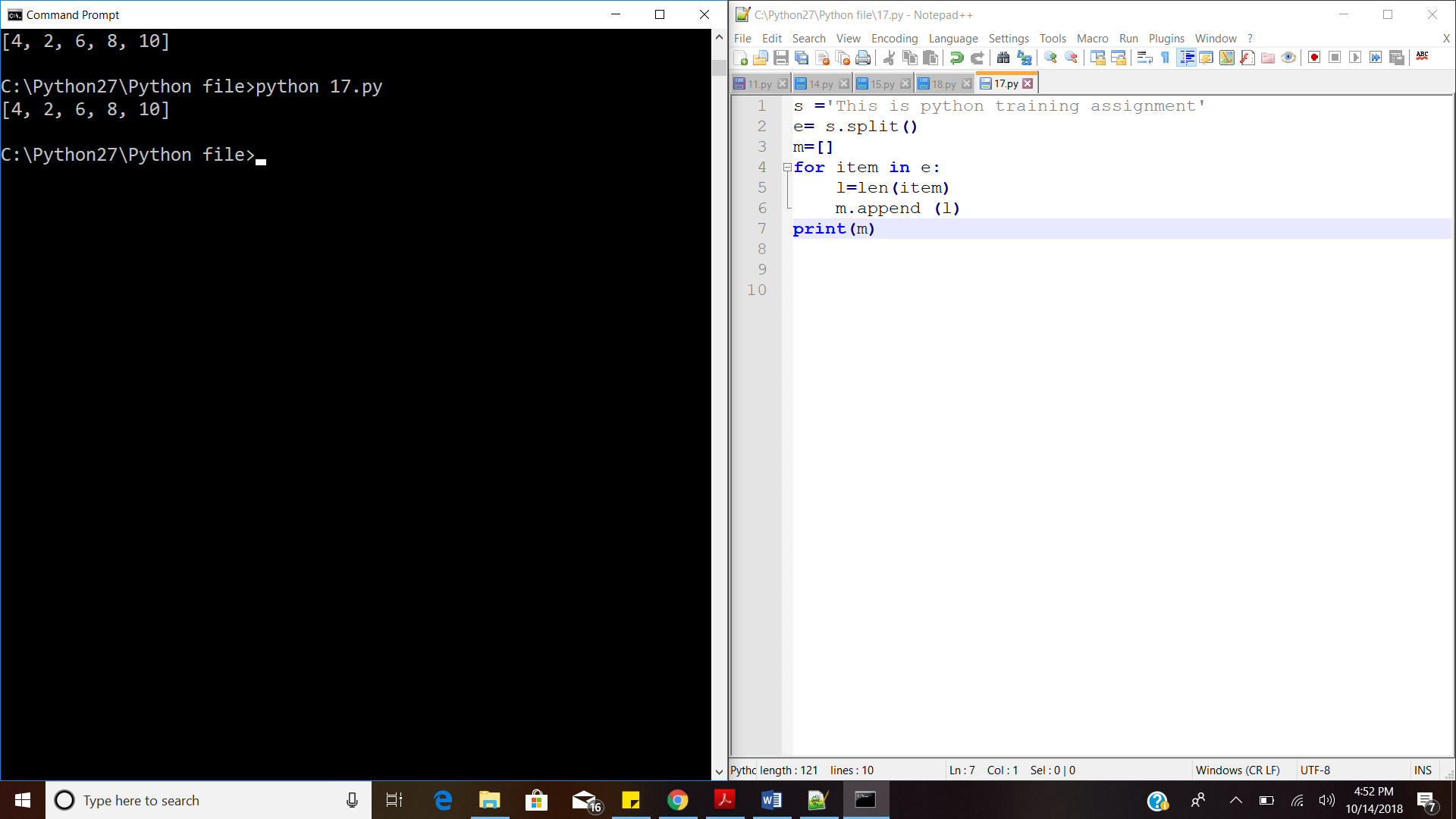
1. **Find out the common character from the two different string using set.**





1. **- Write a program that read the sentence and gives the output as the length of each word in a sentence in the form of a list.**





1. **18 - Write a program to concatenate the following dictionaries to create a new one.**

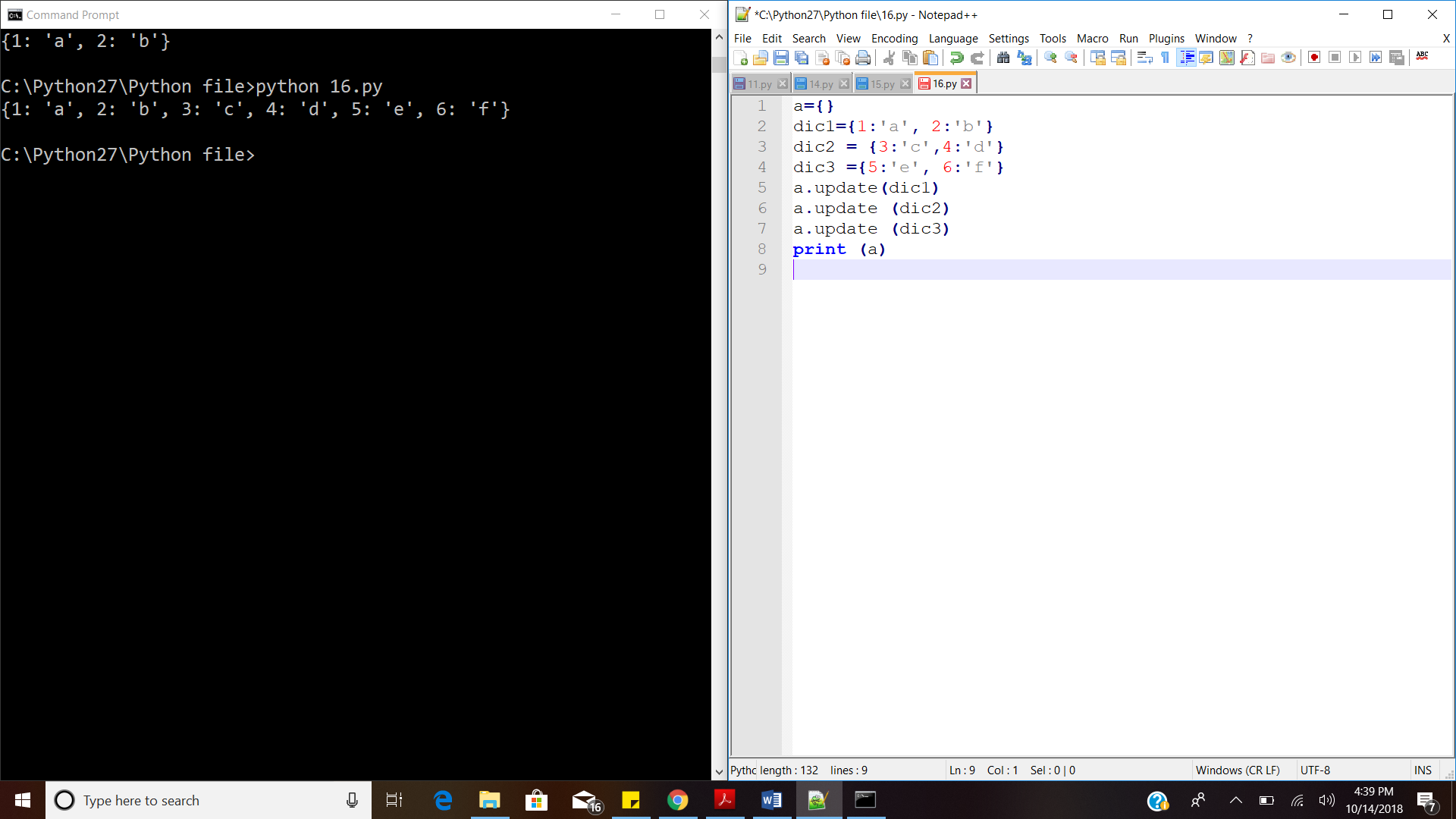
**Sample Dictionary:**

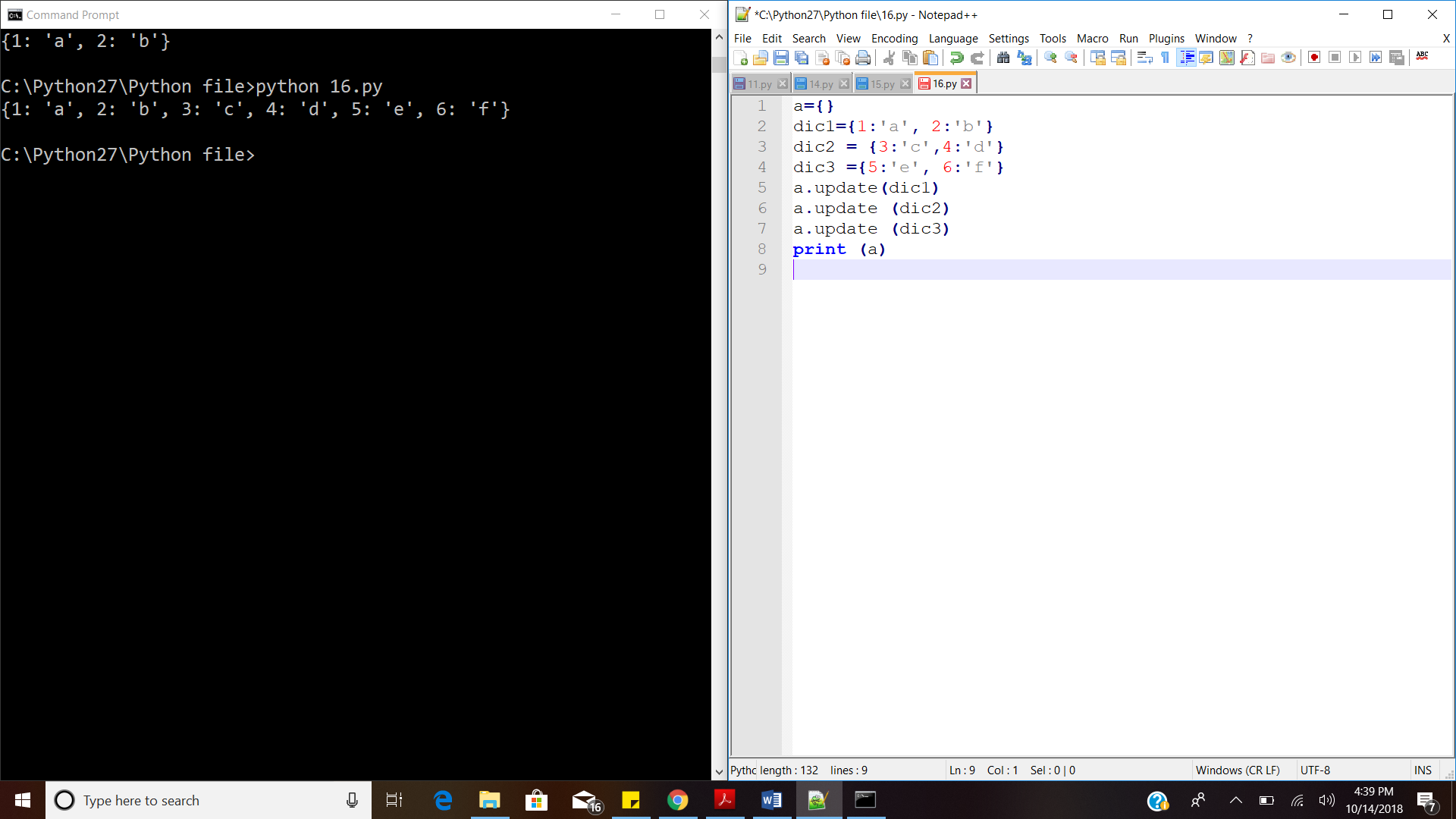
**dic1= {1:’a’, 2:’b’}**

**dic2= {3:’c’, 4:’d’}**

**dic3= {5:’e’, 6:’f’}**

**Expected Result: {1: ‘a’, 2: ‘b’, 3: ‘c’, 4: ‘d’, 5:’e’ , 6: ‘f’}**





1. **What is memory Management in Python?**

Whenever variable in python are assigned with some value, the block of memory is created that holds that value for the variable.

X=10

10

If the value of the variable is changed, the block of memory will be updated with that change.

X=5

5

All the variables having the same value will refer to the same memory location.

1. **What is the difference between range and xrange function.**

**Version 2:** The output of the **range** function is **list**, while the output of the **xrange** function is **object**.

>>> range (5)

[0, 1, 2, 3, 4]

>>> for i in xrange (5):

print (i)

0

1

2

3

4

**Version 3:** The output of the **range** function is **object**, while the **xrange** is **not defined**.

>>> for num in range (5):

... print (num)

...

0

1

2

3

4

>>> for i in xrange (5):

... print (i)

...

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'xrange' is not defined

**Reference**

1. <http://foobarnbaz.com/2012/07/08/understanding-python-variables/>
2. <https://www.geeksforgeeks.org/packing-and-unpacking-arguments-in-python/>