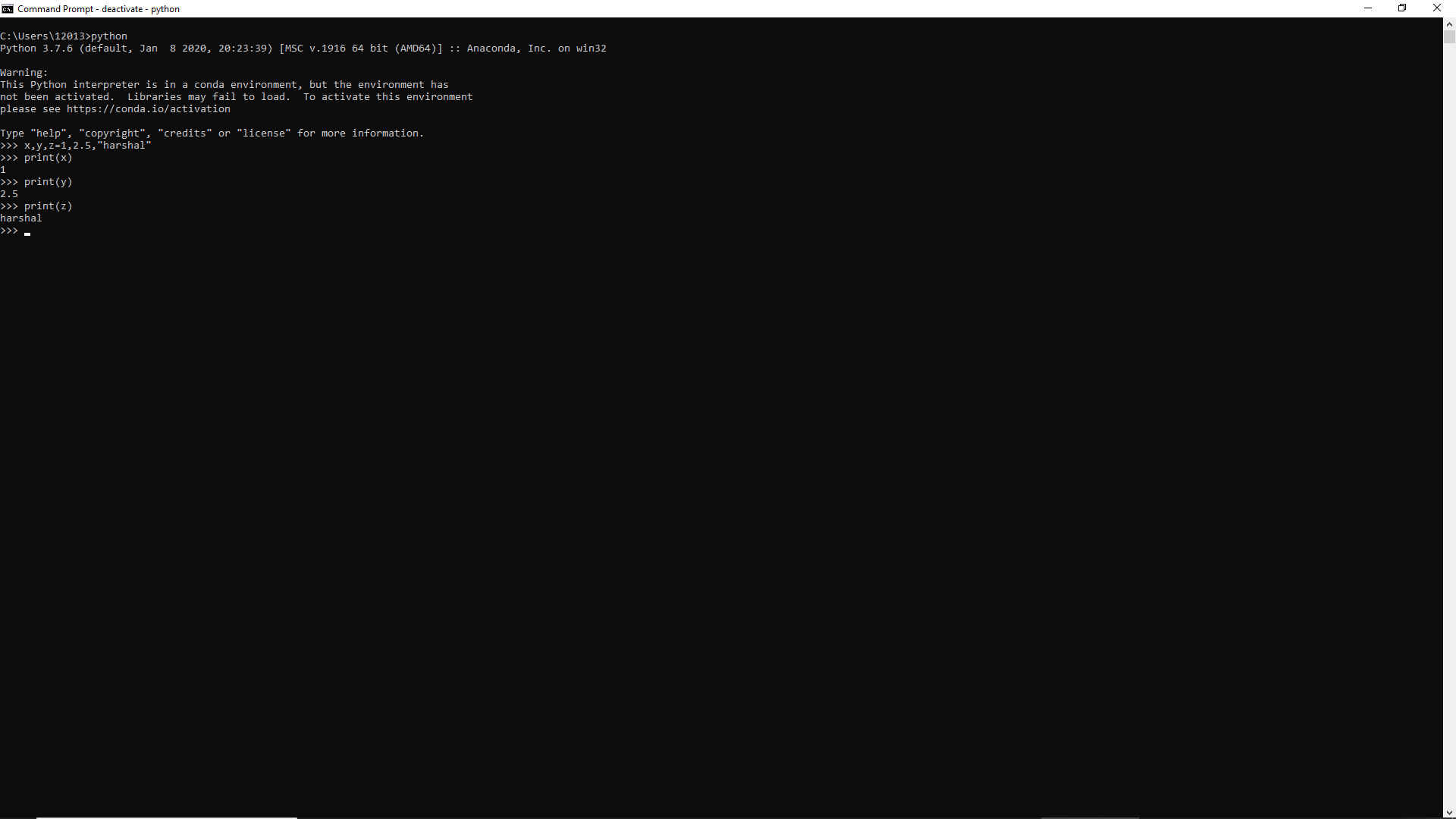
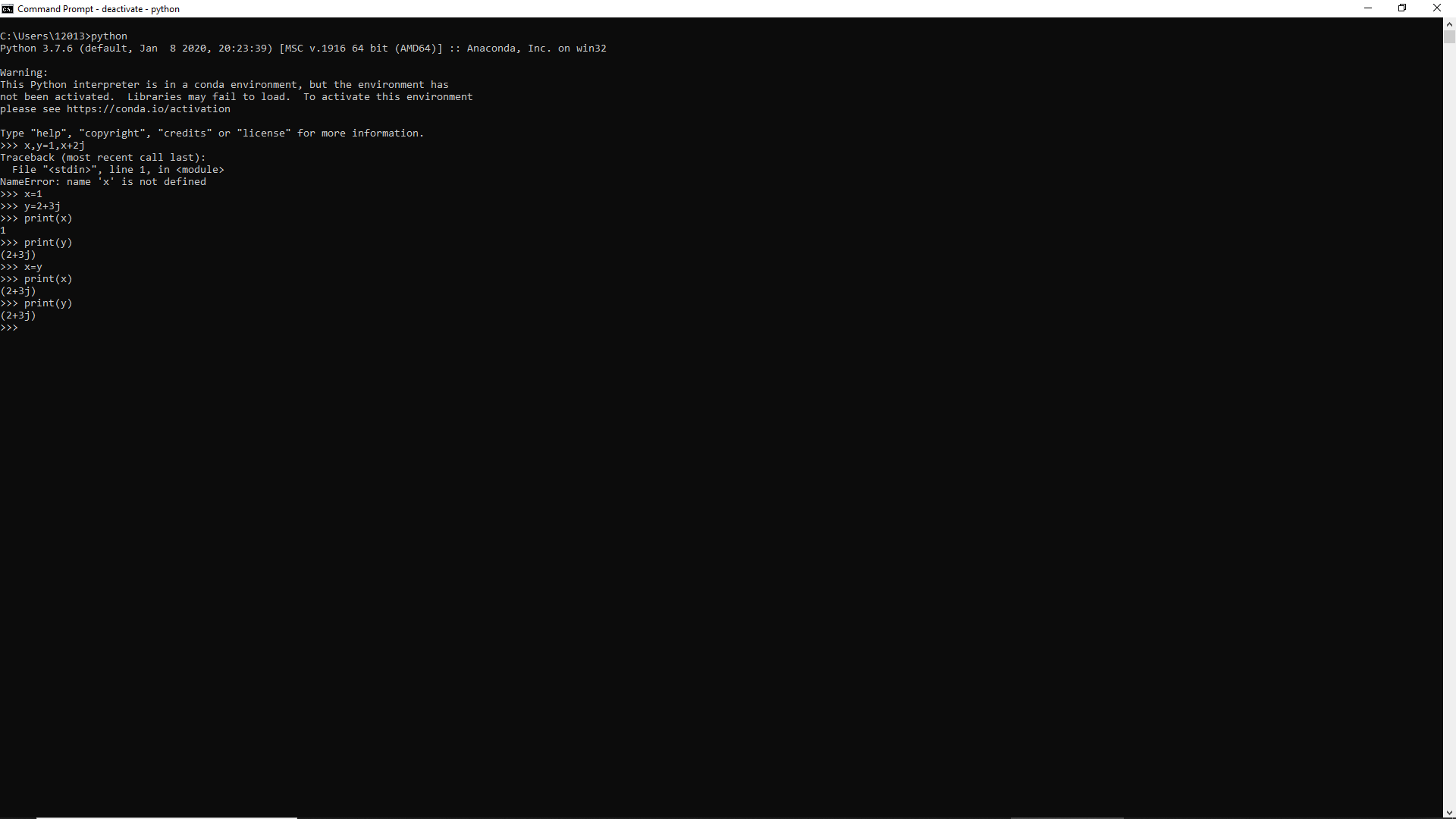
**DAY -11-12 ESTIMATE (2-3 hours)**

**TASK ONE: NUMBERS AND VARIABLES**

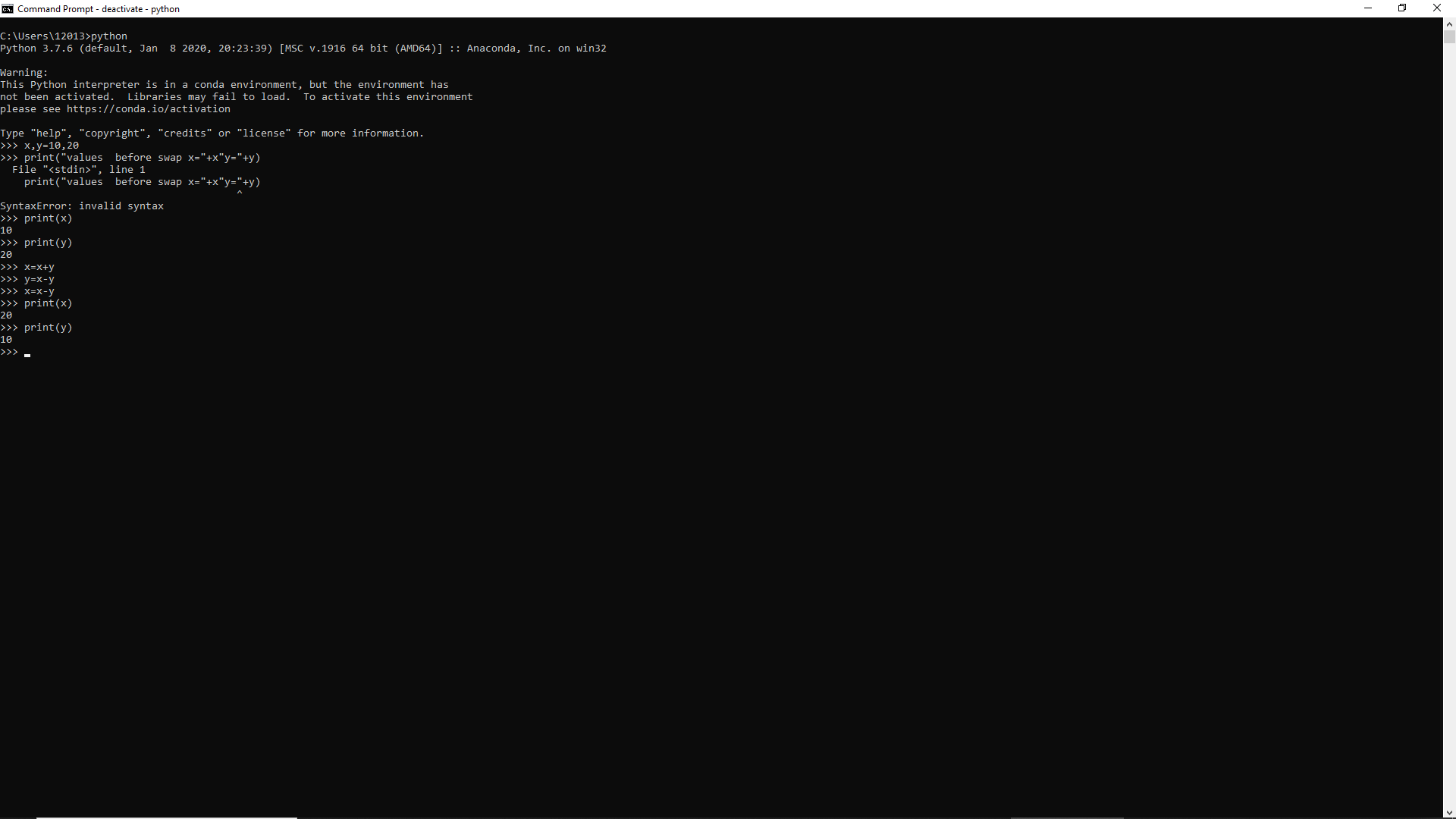
1. Create three variables in a single a line and assign different values to them and make sure their data types are different. Like one is int, another one is float and the last one is a string.

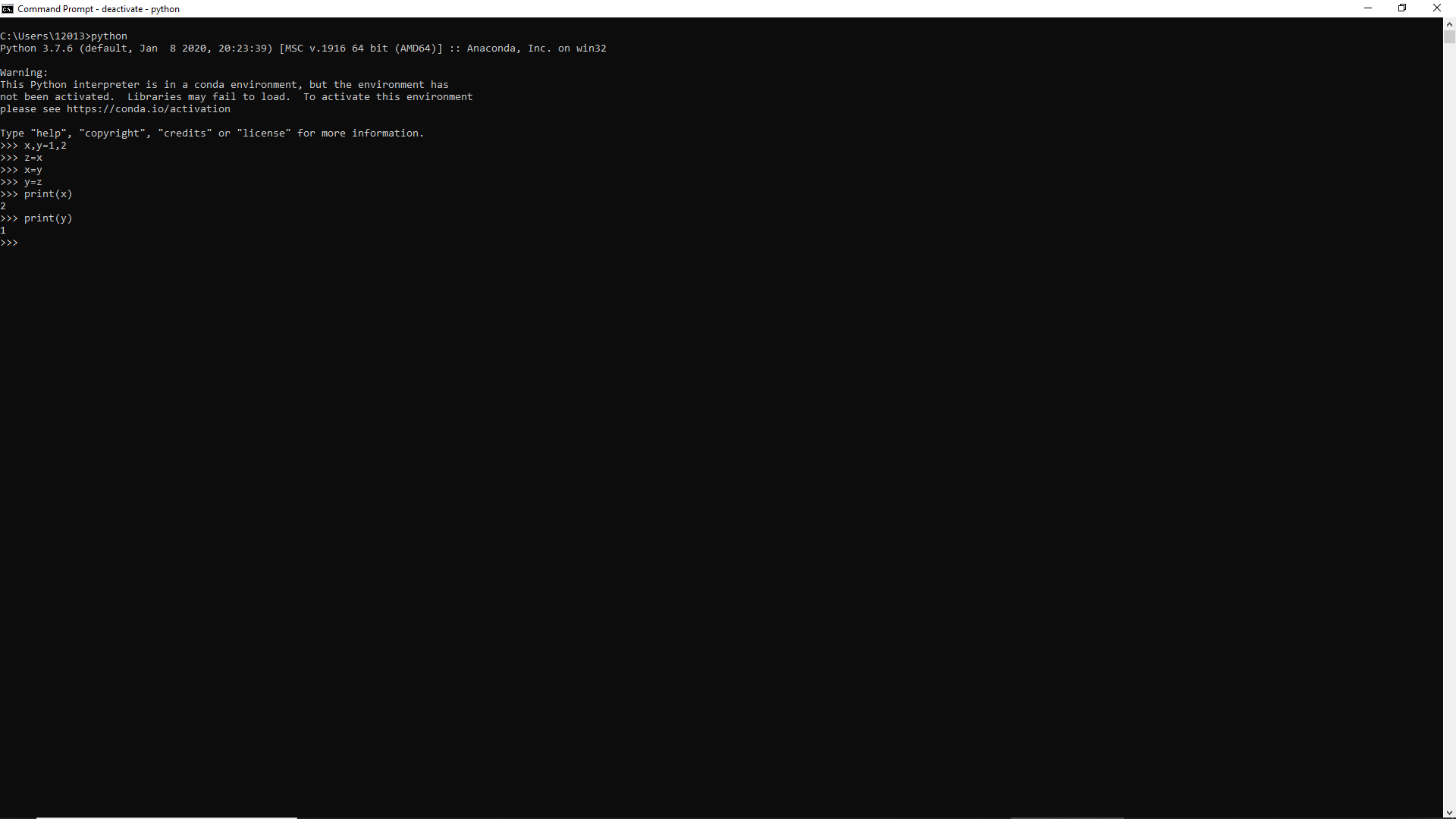


2. Create a variable of value type complex and swap it with another variable whose value is an integer.

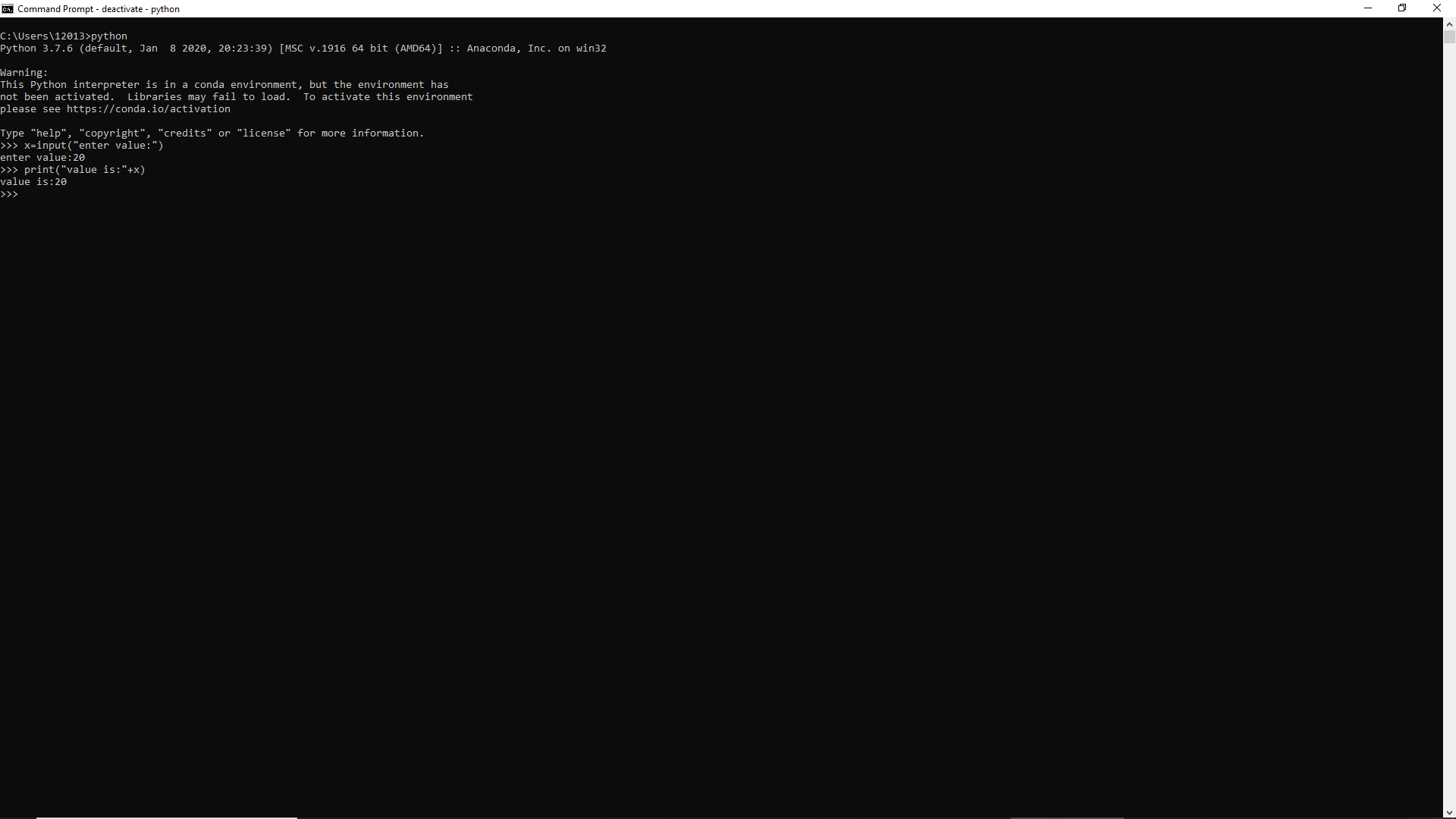


3. Swap two numbers using the third variable as the result name and do the same task without using any third variable.

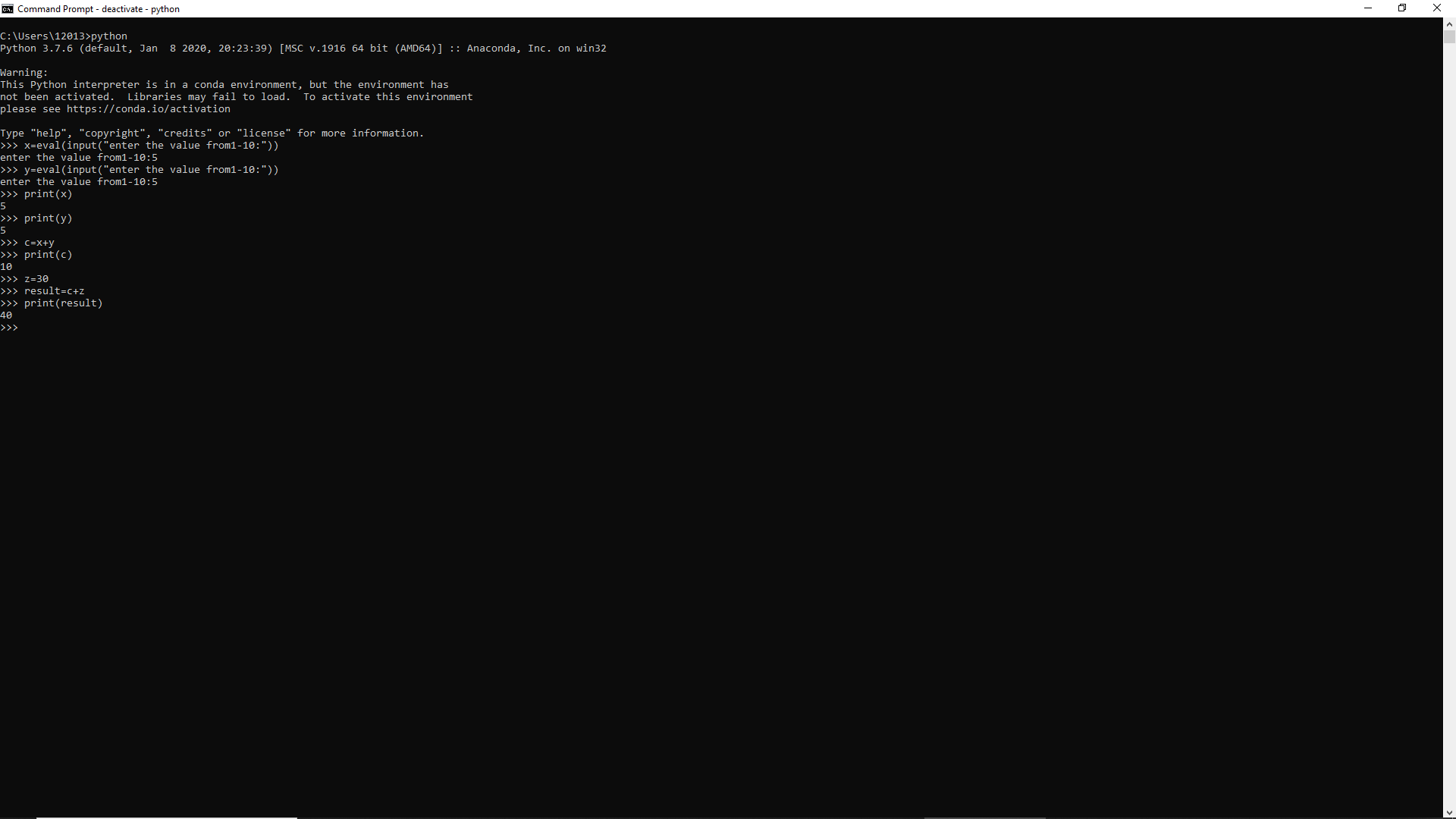




4. Write a program to print the value given by the user by using both Python 2.x and Python 3.x Version.

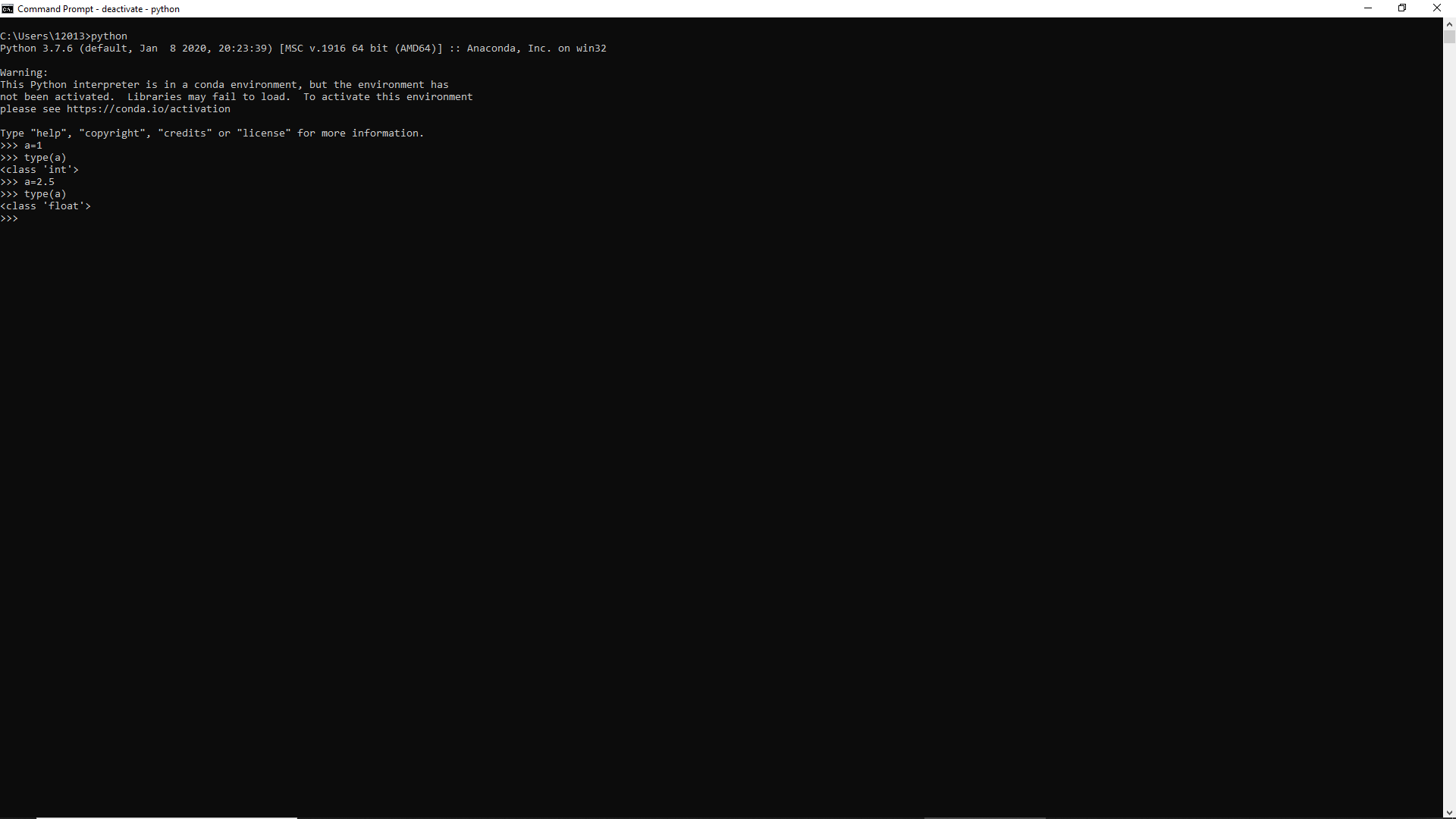


5. Write a program to complete the task given below:

* Ask the user to enter any 2 numbers in between 1-10 and add both of them to another variable call z.
* Use z for adding 30 into it and print the final result by using variable result.

6. Write a program to check the data type of the entered values. HINT: Printed output should say - The input value data type is: int/float/string/etc

7. If one data type value is assigned to ‘a’ variable and then a different data type value is assigned to ‘a’ again. Will it change the value. If Yes then Why?

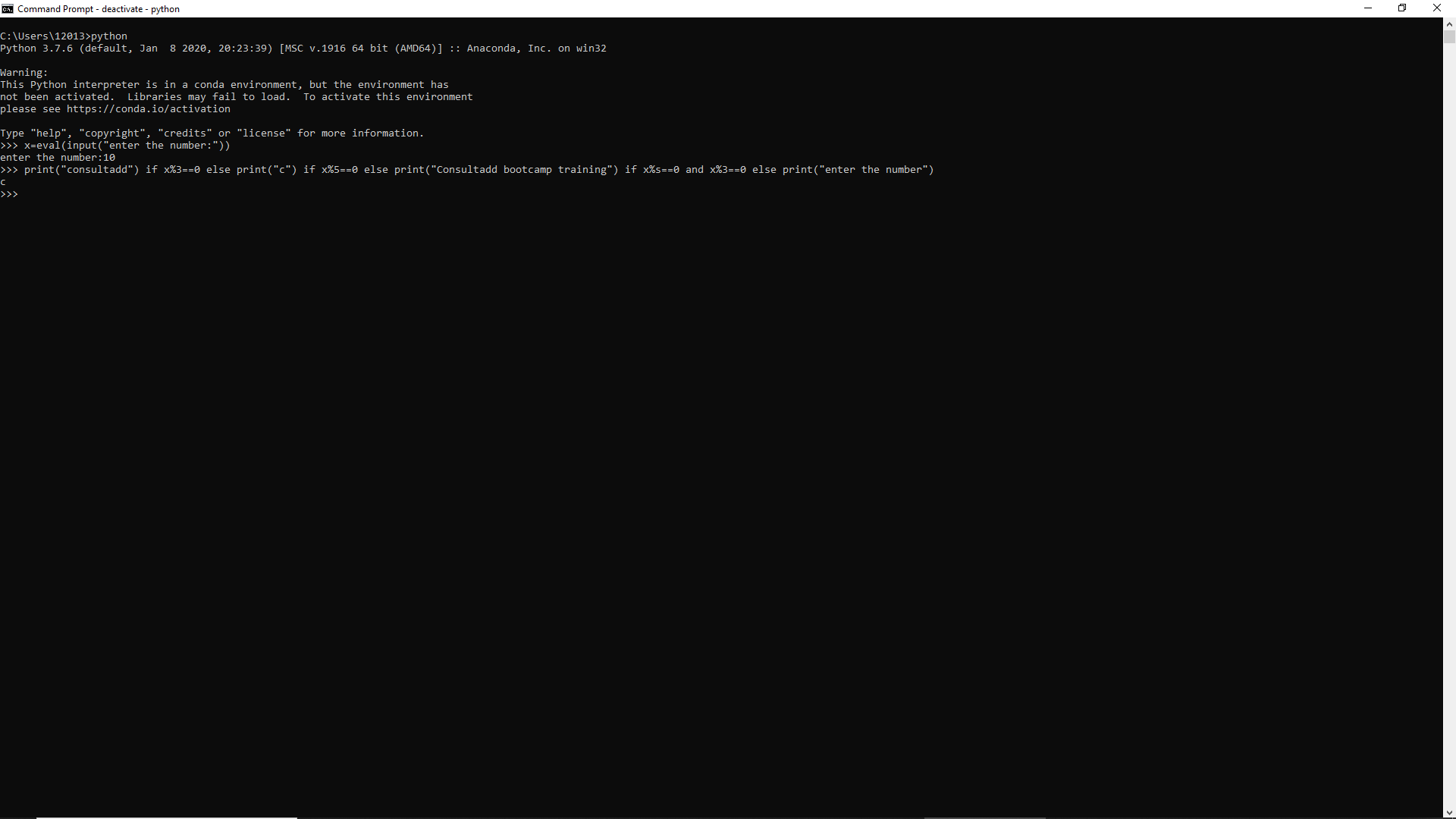


Variables in python they are dynamic you dont have to assign datatype while declearing then like other languages (ex: int a)

**TASK TWO: OPERATORS AND DECISION MAKING STATEMENT**

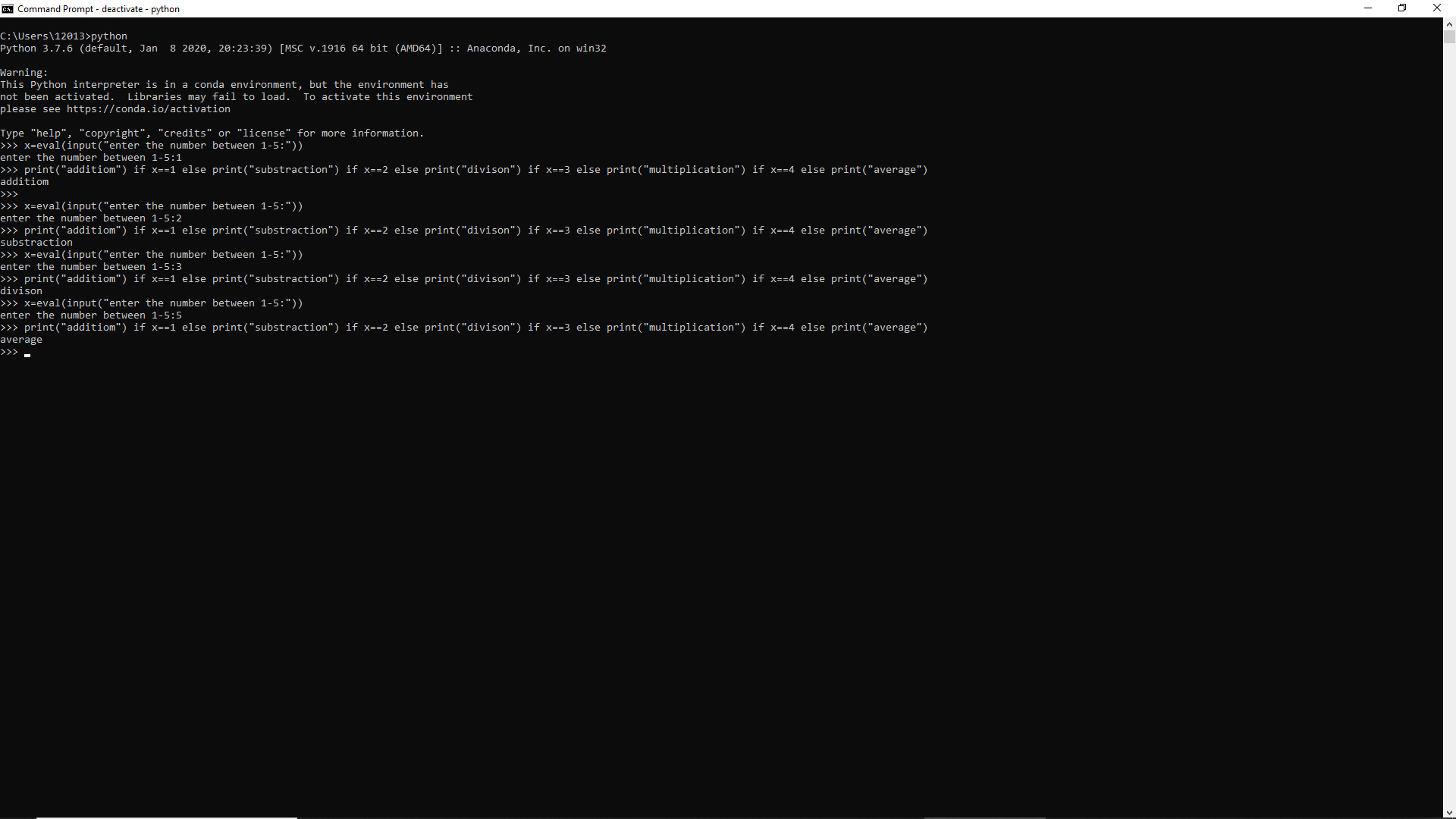
1. Write a program in Python to perform the following operation:

* If a number is divisible by 3 it should print “Consultadd” as a string
* If a number is divisible by 5 it should print “c” as a string
* If a number is divisible by both 3 and 5 it should print “Consultadd Python Training” as a string.

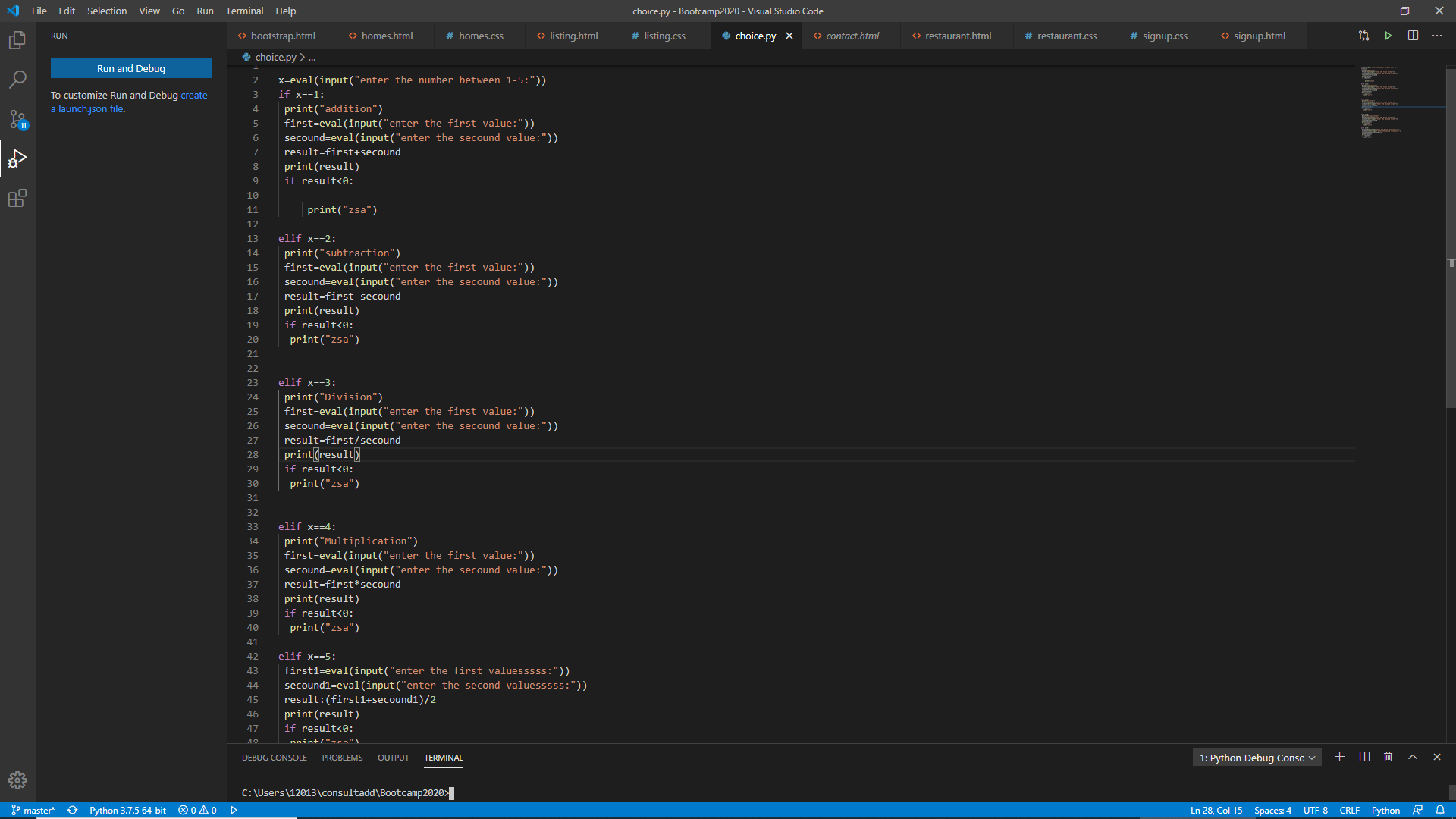


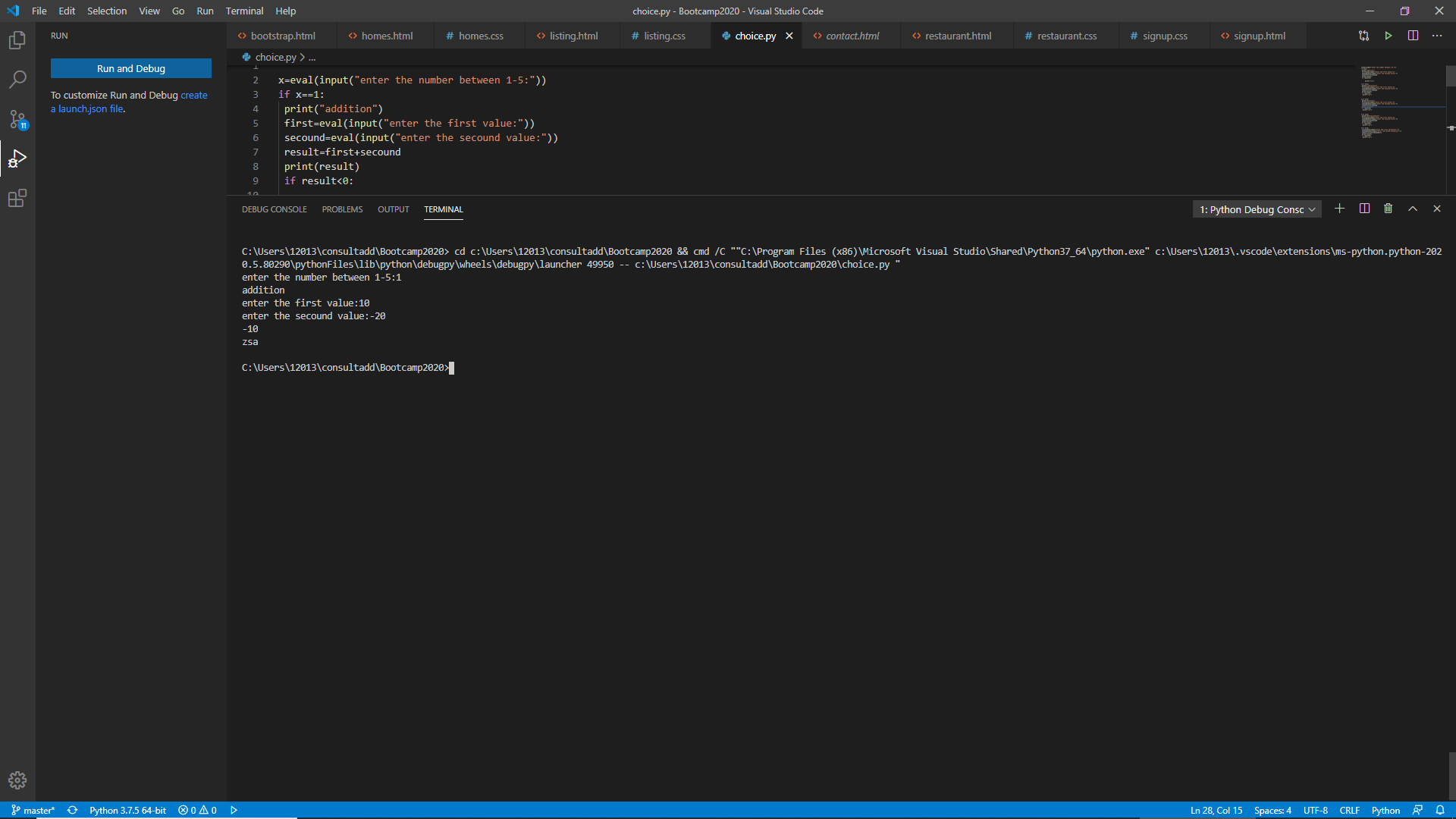
2. Write a program in Python to perform the following operator based task:

* Ask user to choose the following option first:
  + If User Enter 1 - Addition
  + If User Enter 2 - Subtraction
  + If User Enter 3 - Division
  + If USer Enter 4 - Multiplication
  + If User Enter 5 - Average

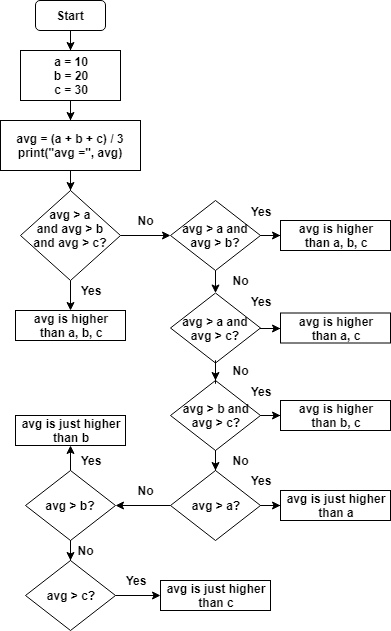


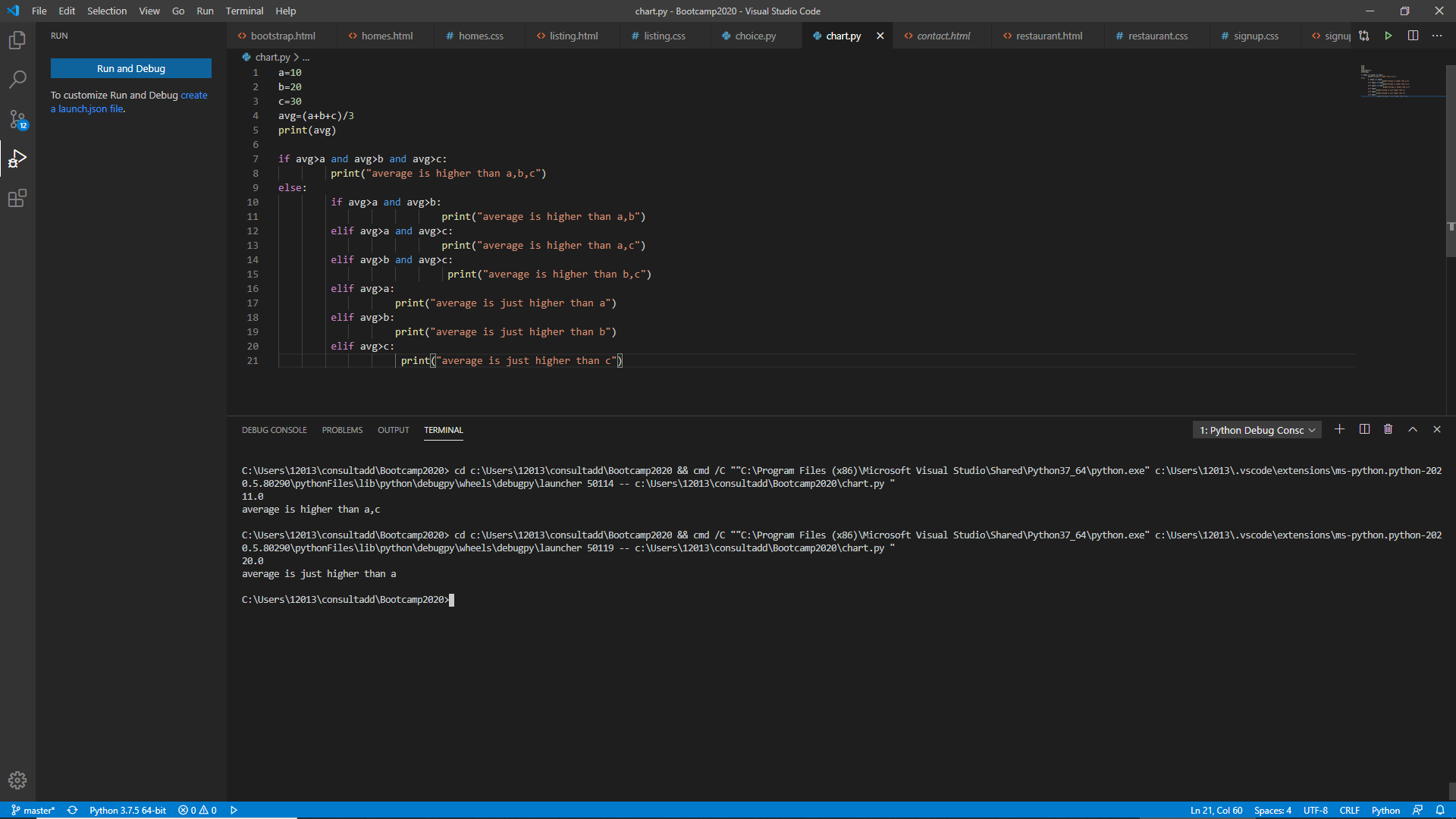
* Ask user to enter the 2 numbers in a variable for first and second for the first 4 options mentioned above.
* Ask user to enter two more numbers as first1 and second2 for calculating the average as soon as user choose an option 5.
* At the end if the answer of any operation is Negative print a statement saying “zsa”
* NOTE: At a time user can perform one action at a time.





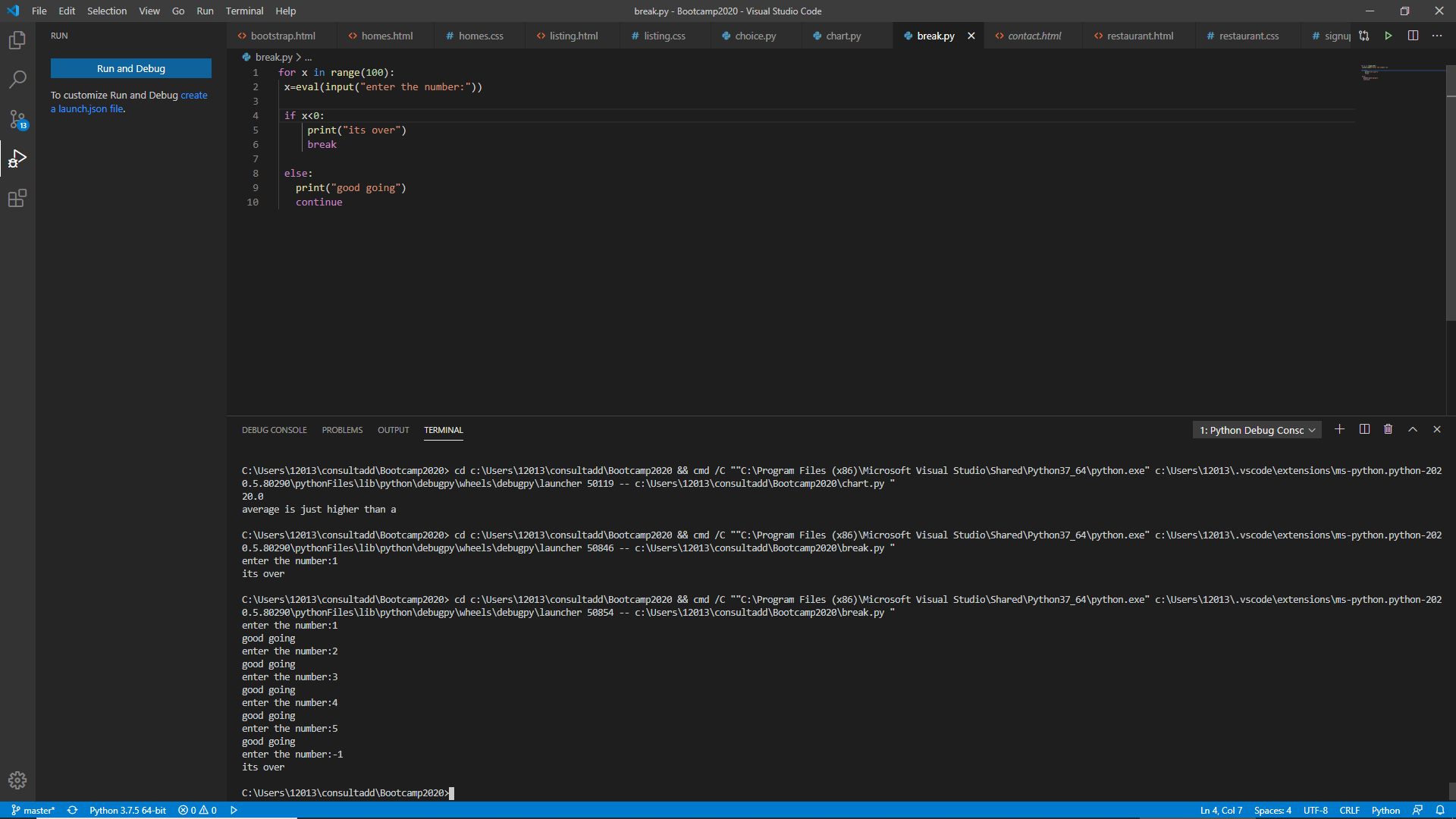
3. Write a program in Python to implement the given flowchart:



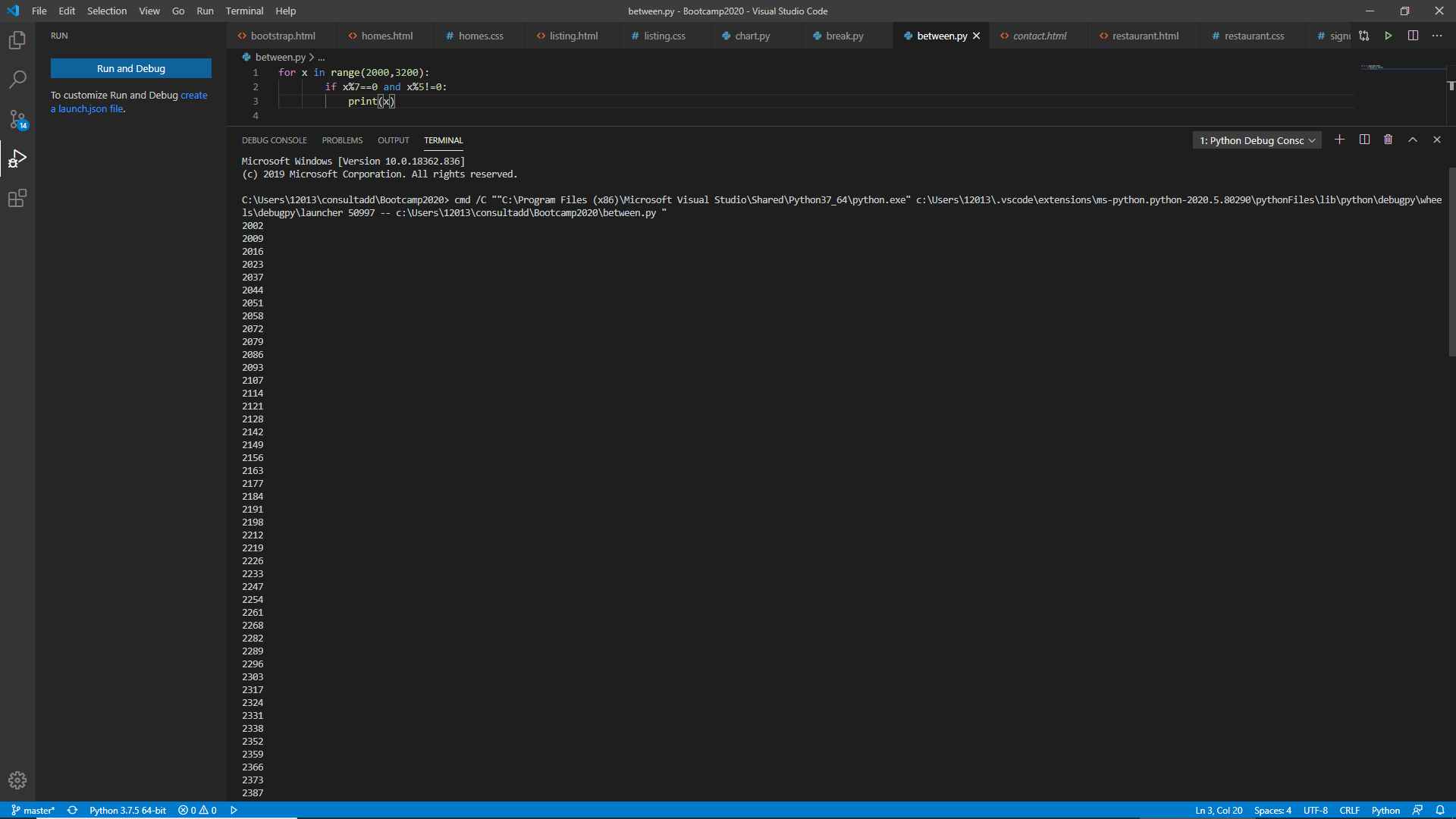


4. Write a program in Python to break and continue if the following cases occurs:

* If user enters a negative number just break the loop and print “It’s Over”
* If user enters a positive number just continue in the loop and print “Good Going”



5. Write a program in Python which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200.

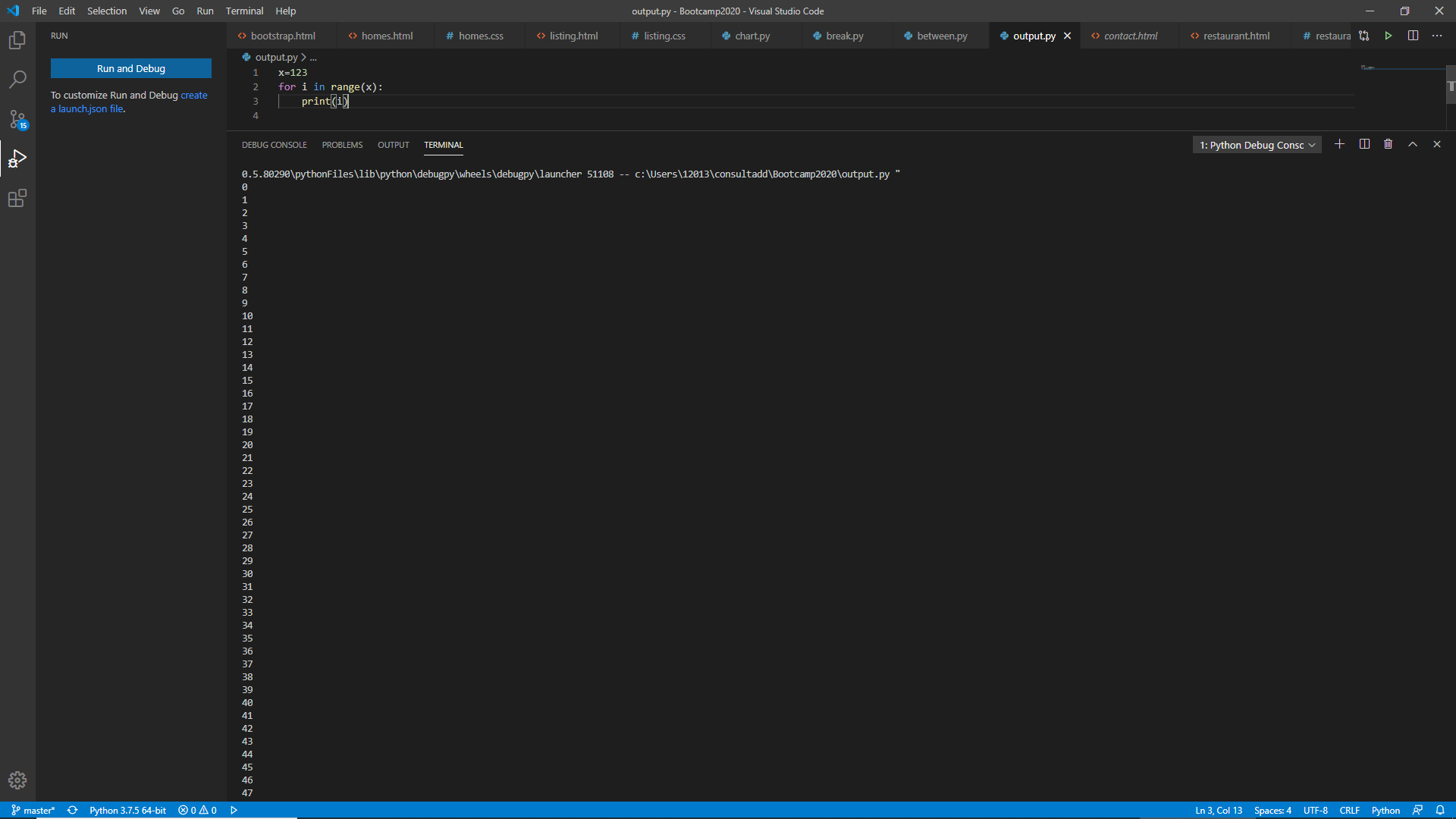


6. What is the output of the following code examples?

* x=123

for i in x:

print(i)



* i = 0

while i < 5:

print(i)

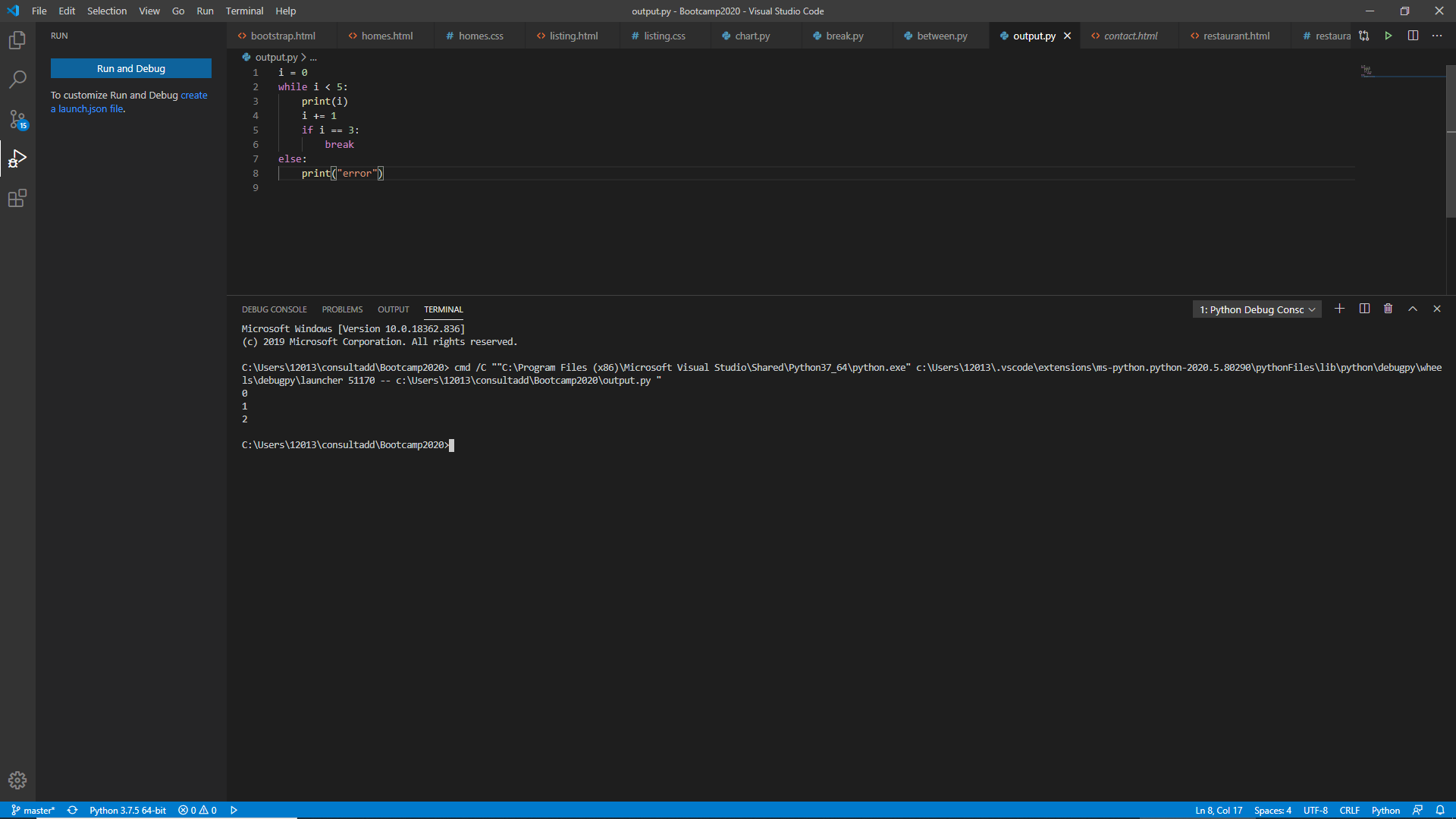
i += 1

if i == 3:

break

else:

print(“error”)



* count = 0

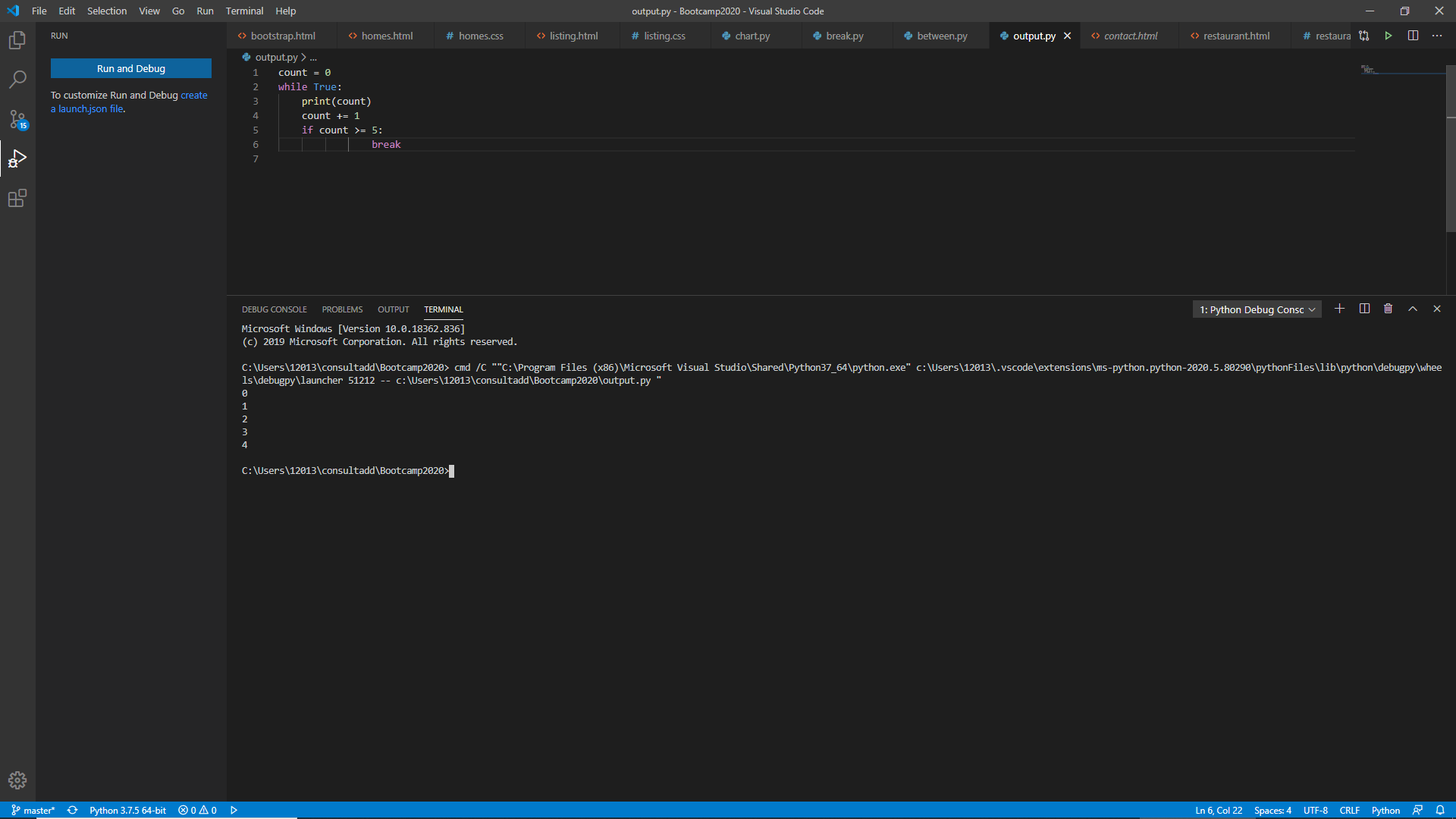
while True:

print(count)

count += 1

if count >= 5:

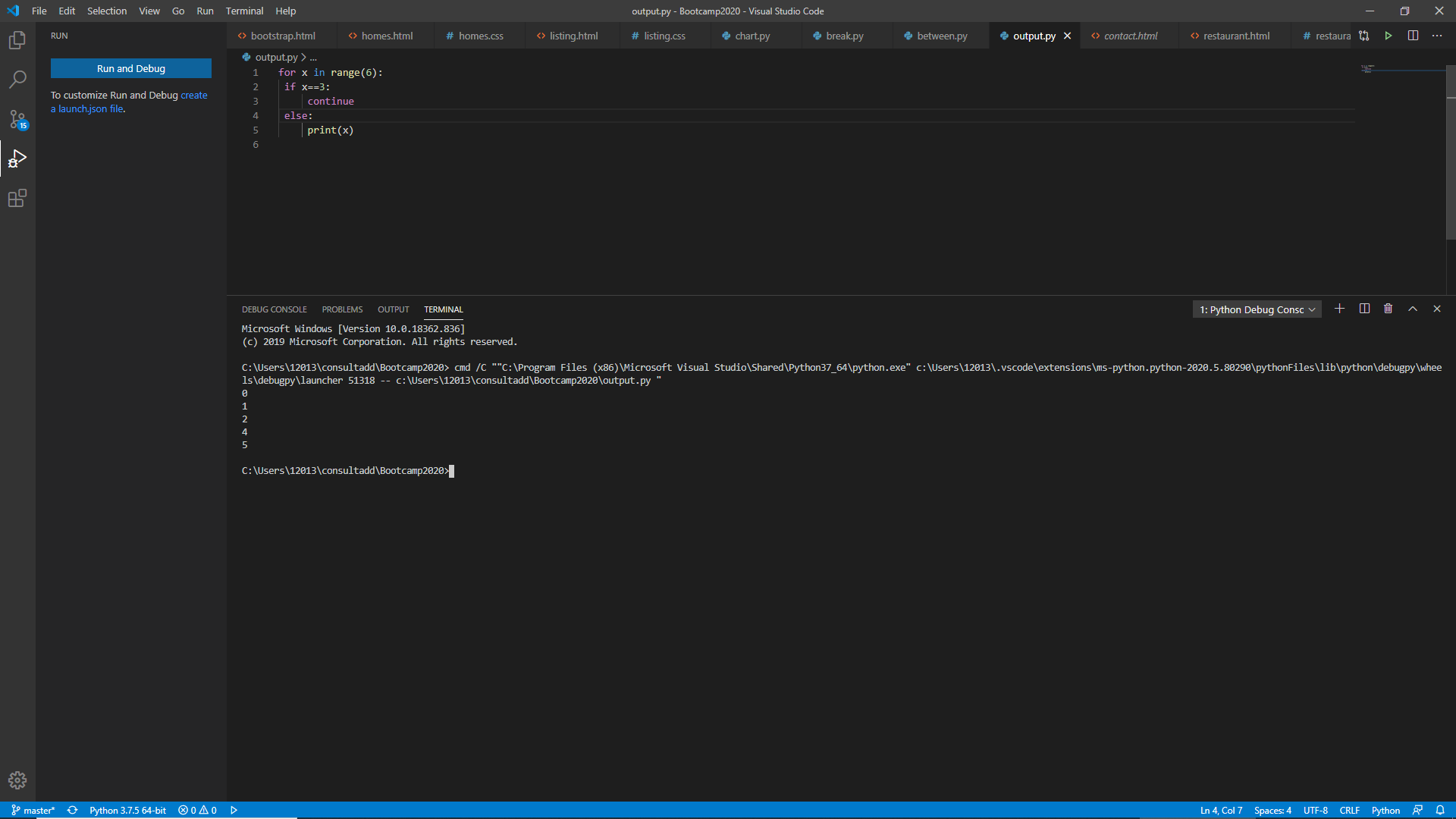
Break



7. Write a program that prints all the numbers from 0 to 6 except 3 and 6.

Expected output: 0 1 2 4 5

Note: Use ‘continue’ statement

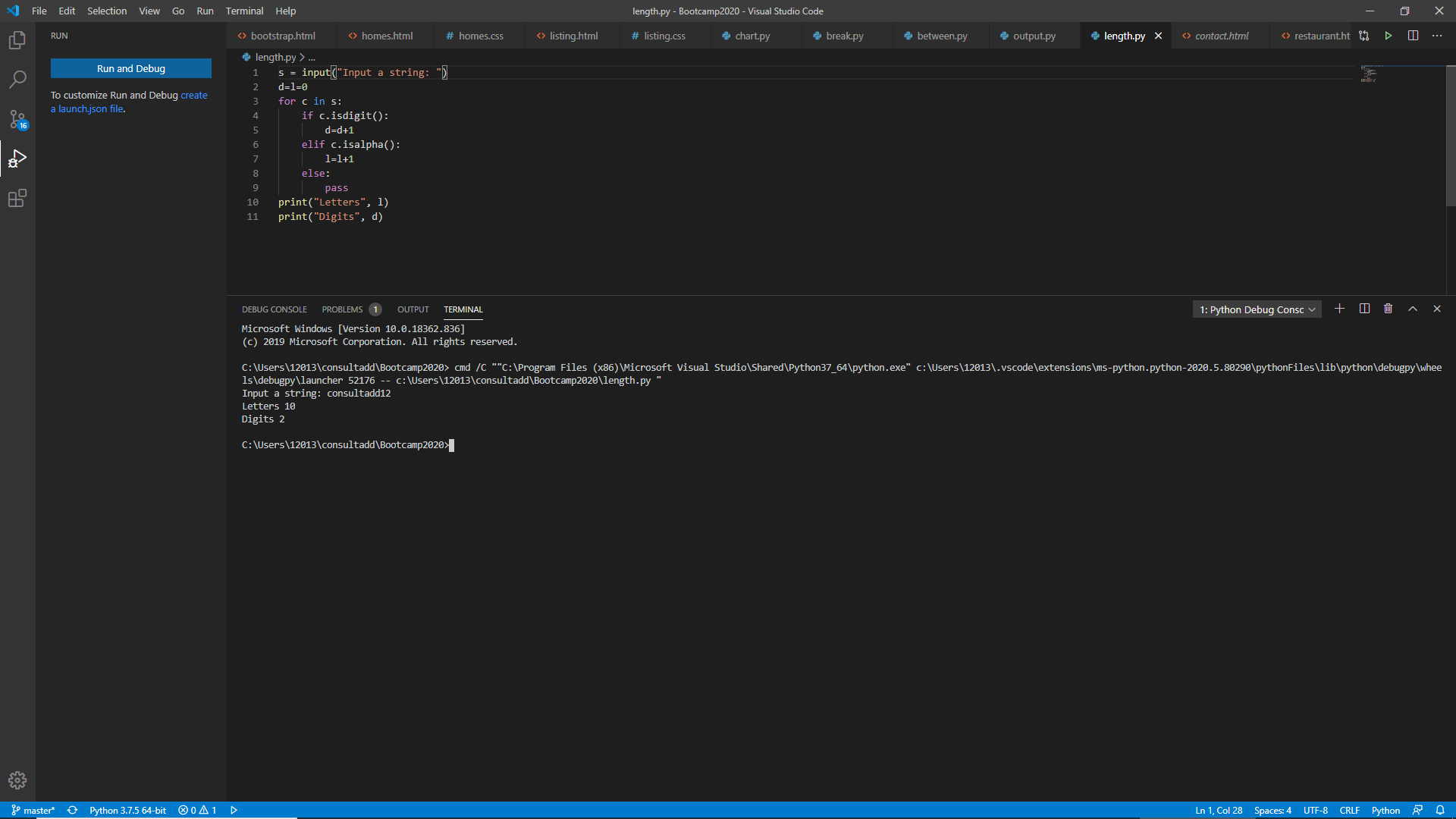


8. Write a program that accepts a string as an input from user and calculate the number of digits and letters.

Expected output: consul12

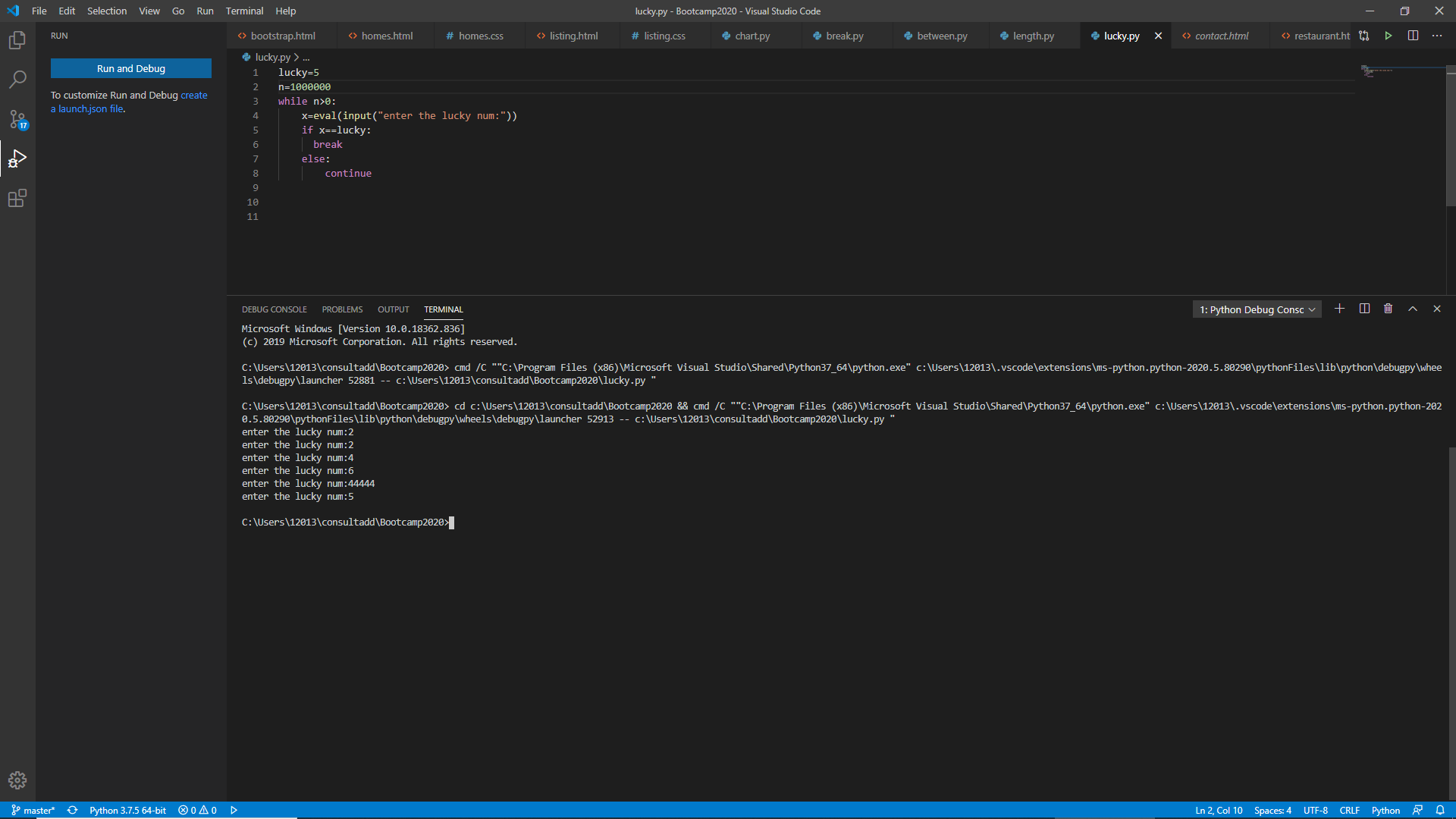
Letters 6

Digits 2

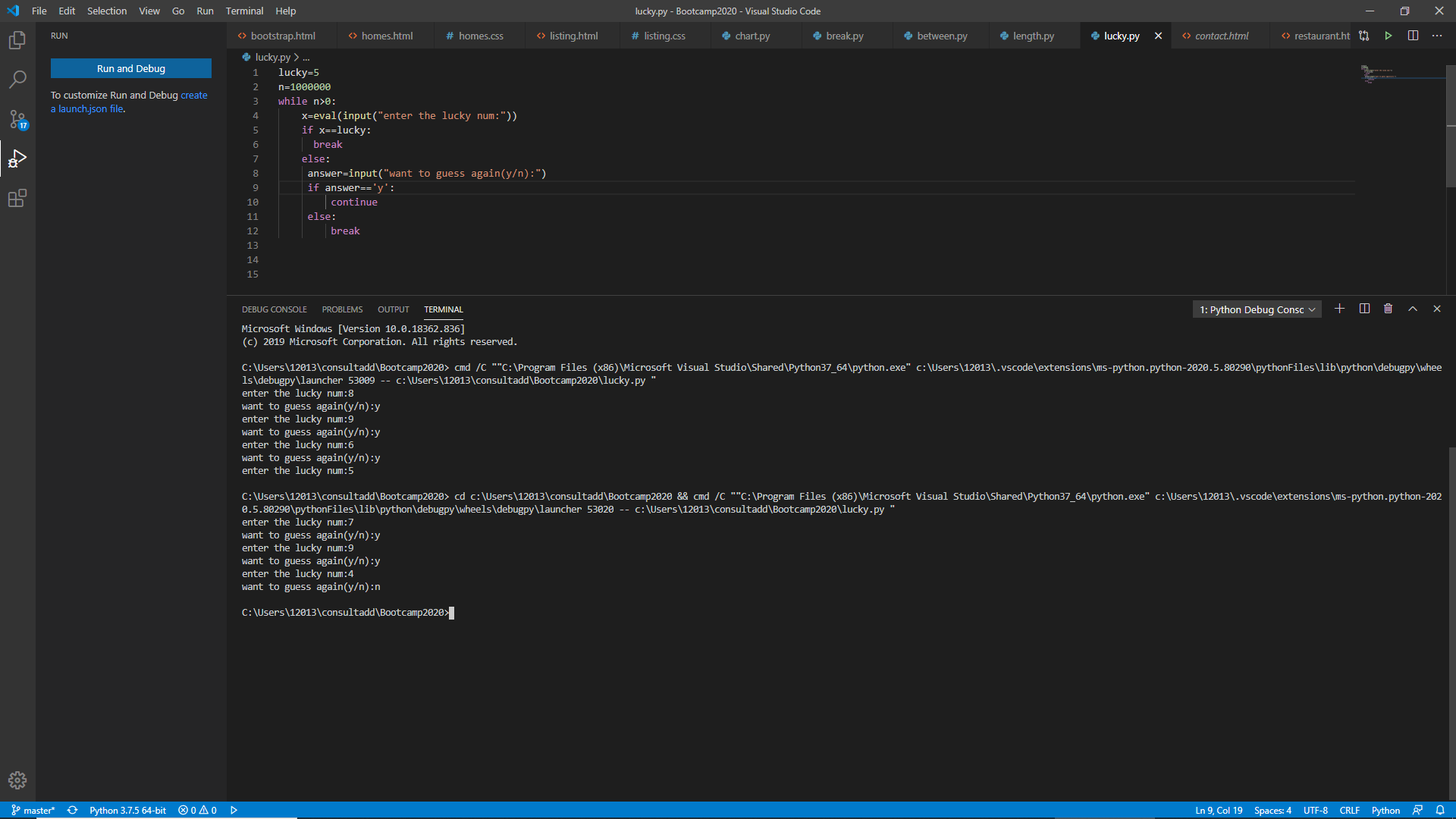


9. Read the two parts of the question below:

* Write a program such that it asks users to “guess the lucky number”. If the correct number is guessed the program stops, otherwise it continues forever.



* Modify the program so that it asks users whether they want to guess again each time. Use two variables, ‘number’ for the number and ‘answer’ for the answer to the question whether they want to continue guessing. The program stops if the user guesses the correct number or answers “no”. ( The program continues as long as a user has not answered “no” and has not guessed the correct number)



10. Write a program that asks five times to guess the lucky number. Use a while loop and a counter, such as

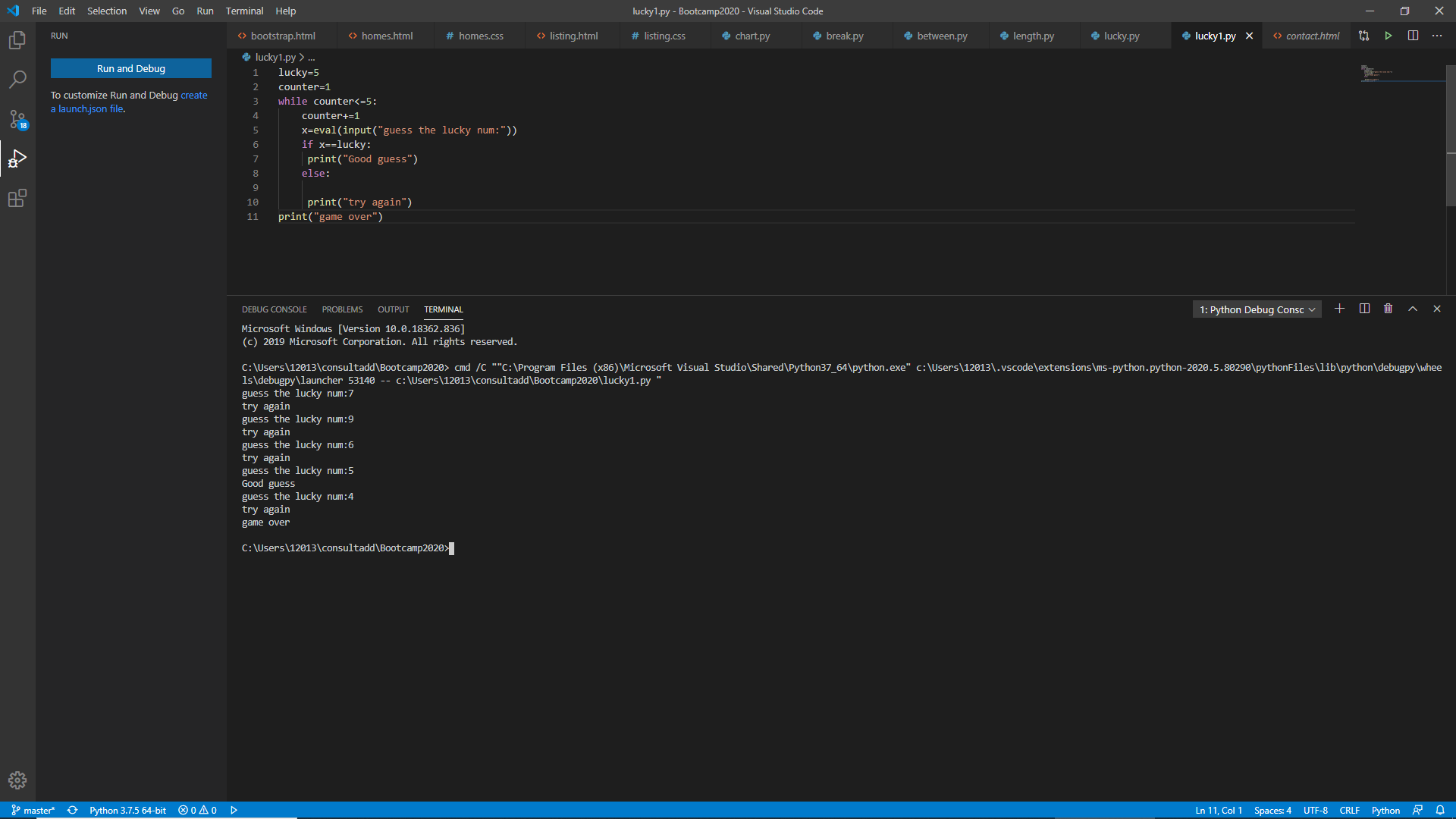
counter=1

While counter <= 5:

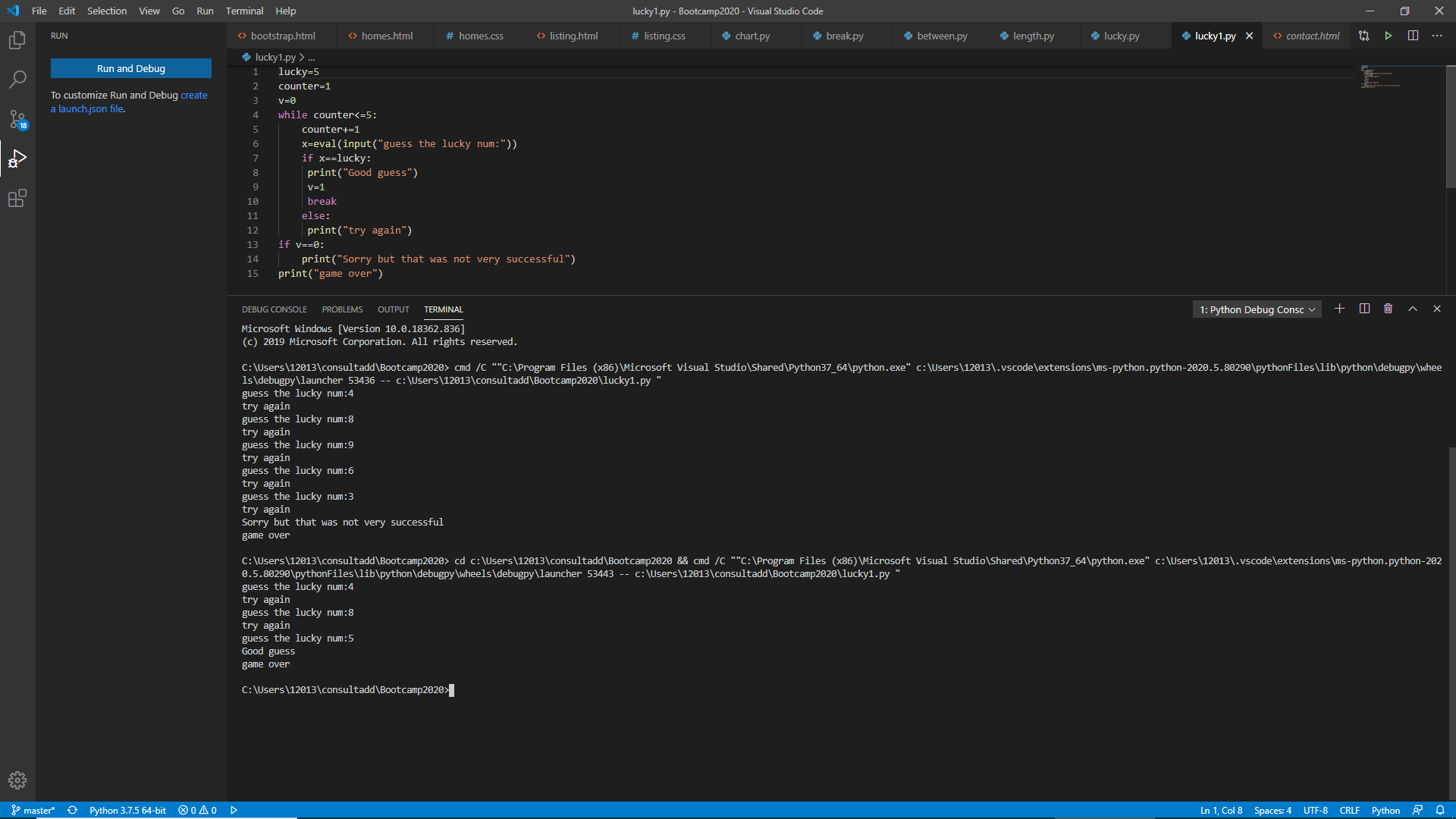
print(“Type in the”, counter, “number”

counter=counter+1

The program asks for five guesses (no matter whether the correct number was guessed or not). If the correct number is guessed, the program outputs “Good guess!”, otherwise it outputs “Try again!”. After the fifth guess it stops and prints “Game over!”.

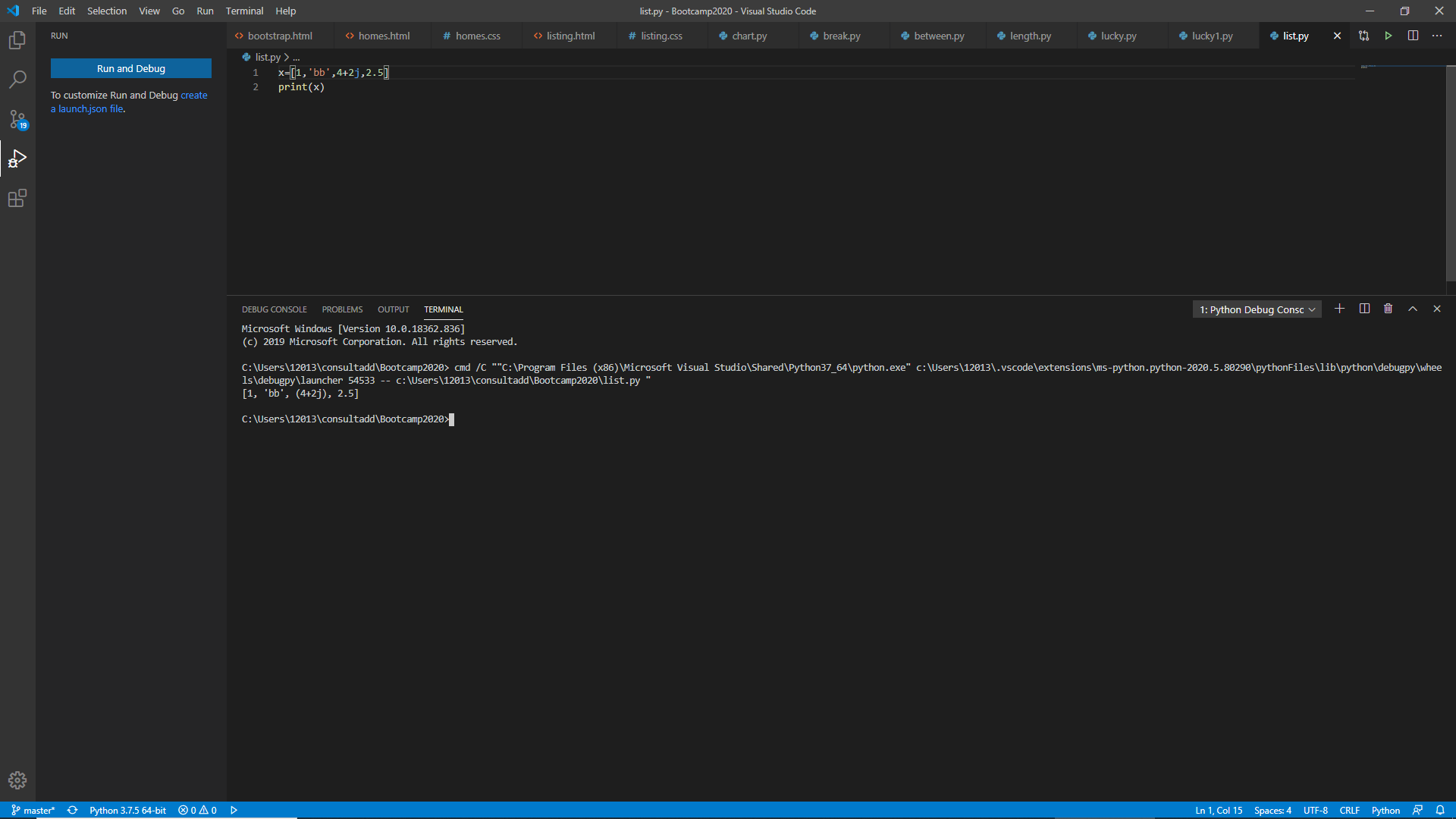


11. In the previous question, insert “break” after the “Good guess!” print statement. “break” will terminate the while loop so that users do not have to continue guessing after they found the number. If the user does not guess the number at all, print “Sorry but that was not very successful”.



**TASK THREE: DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.



2. Create a list of size 5 and execute the slicing structure

3. Write a program to get the sum and multiply of all the items in a given list.

4. Find the largest and smallest number from a given list.

5. Create a new list which contains the specified numbers after removing the even numbers from a predefined list.

6. Create a list of first and last 5 elements where the values are square of numbers between 1 and 30 (both included).

7. Write a program to replace the last element in a list with another list.

Sample data: [[1,3,5,7,9,10],[2,4,6,8]]

Expected output: [1,3,5,7,9,2,4,6,8]

8. Create a new dictionary by concatenating the following two dictionaries:

a={1:10,2:20}

b={3:30,4:40}

Expected Result: {1:10,2:20,3:30,4:40}

9. Create a dictionary that contains a number (between 1 and n) in the form(x,x\*x).

Sample data (n=5)

Expected Output: {1:1,2:4,3:9,4:16,5:25}

10. Write a program which accepts a sequence of comma-separated numbers from 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

The output should be:

[‘34’,’67’,’55’,’33’,’12’,’98’]

(‘34’,’67’,’55’,’33’,’12’,’98’)

**WEEKEND ACTIVITY ON DATA STRUCTURES**

1. Create a list of the 10 elements of four different types of Data Types like int, string, complex and float.

2. Create a list of size 5 and execute the slicing structure

3. Create a list of given structure and run

**x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]**

* Access list [1, 2, 3, 4]
* Access list [600, 700]
* Access list [100, 300, 500, 600, 800]
* Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]
* Access list [10]
* Access list [ ]

4. Create a list of thousand number using range and xrange and see the difference between each other.

5. How Tuple is beneficial as compare to the list?

6. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

7. Write a program in Python to reverse a string and print only the vowel alphabet if exist in the string with their index.

8. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which has even length of word.

9. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

**x=[1,2,3,4,5,6,7,8,9,-1]**

10. Write a program in Python to complete the following task:

* Create two different list as in even\_list and odd\_list
* Ask user to enter the number in the range of 1,50 and make sure if the entered number is even append it to the even\_list and if the entered number is odd append it to the odd list.
* Keep that in mind you can only add 5 items in each list
* Make sure once you entered the total 5 element calculate the sum of the list and return the maximum out of the list.

11. Write a program to find out the occurrence of a specific word from an alphanumeric statement. **Example:** 12abcbacbaba344ab

**Output:** a=5 b=5 c=2 make sure you should avoid the numbers in you logic

12. Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).