Ralph Gene D. Clado CY Level 4

Intro to Programming:

Assessment 1:

Exercise 1:

word1 = Coding

word2 = is

word3 = Cool

# Use string concatenation to combine the variables and print the phrase

\_\_\_\_\_(\_\_\_\_\_\_\_\_\_)

Print(word1 + " " + word2 + " " + word3)

Exercise 2:

1. Declare a variable and initialize it with the integer value 8.
2. Declare a second variable and initialize it with the integer value 10.
3. Declare a third variable that stores the sum of first two numbers.
4. Print the value of the sum to the console.

Declare variables

num1 = 8

num2 = 10

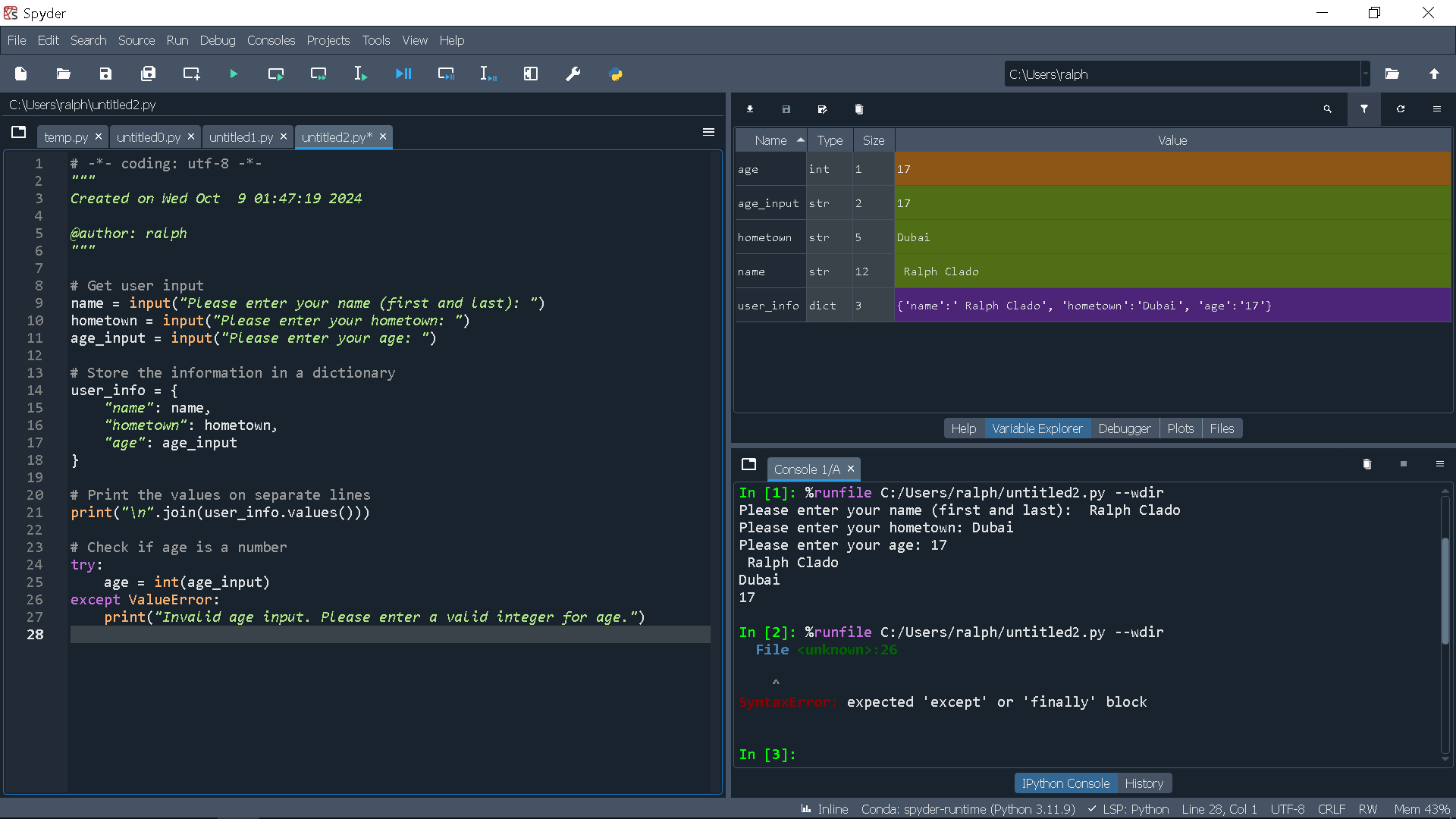
Calculate the sum

sum\_value = num1 + num2

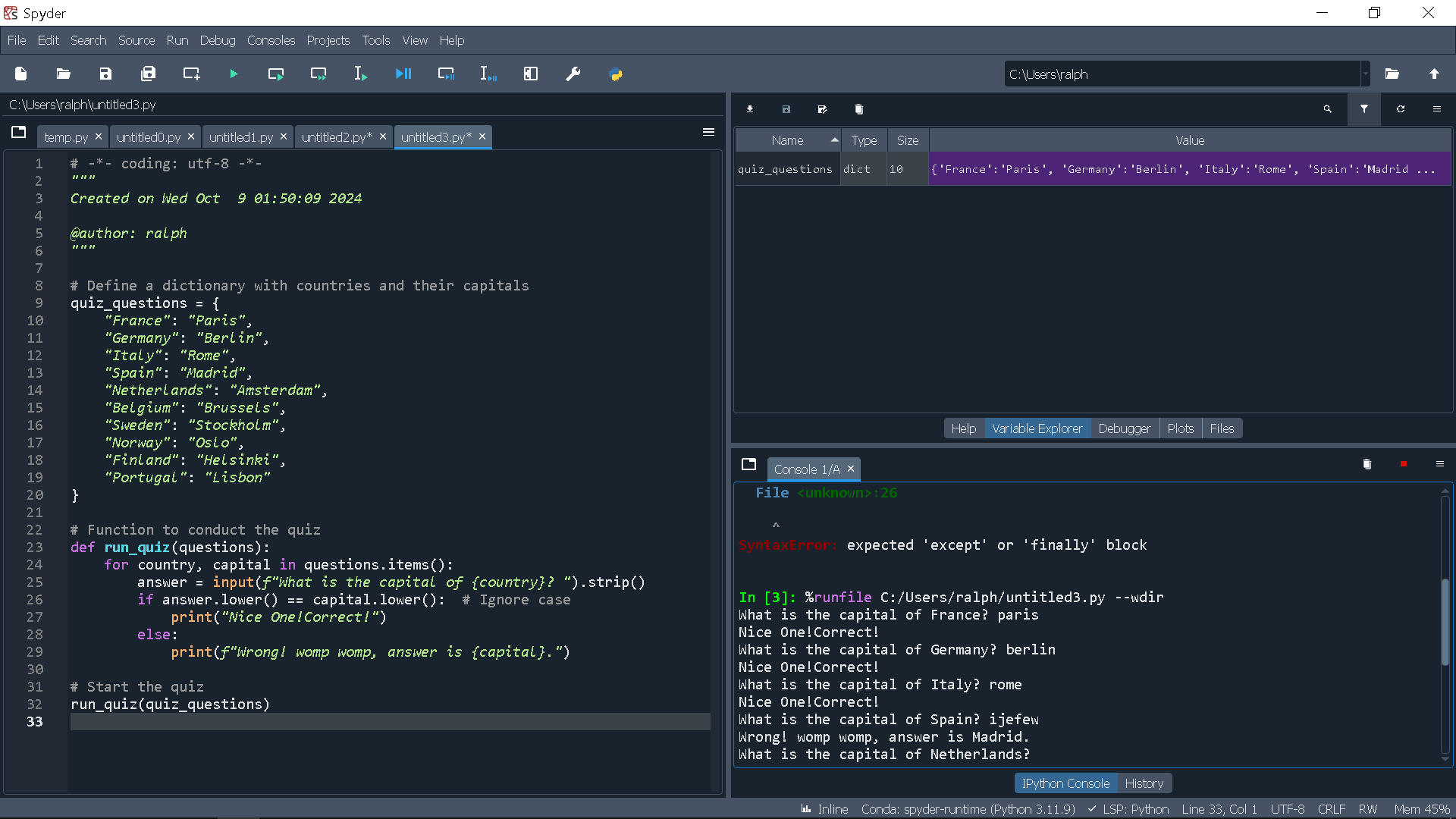
Print the sum

print(sum\_value)

Output: 18



Exercise 4:



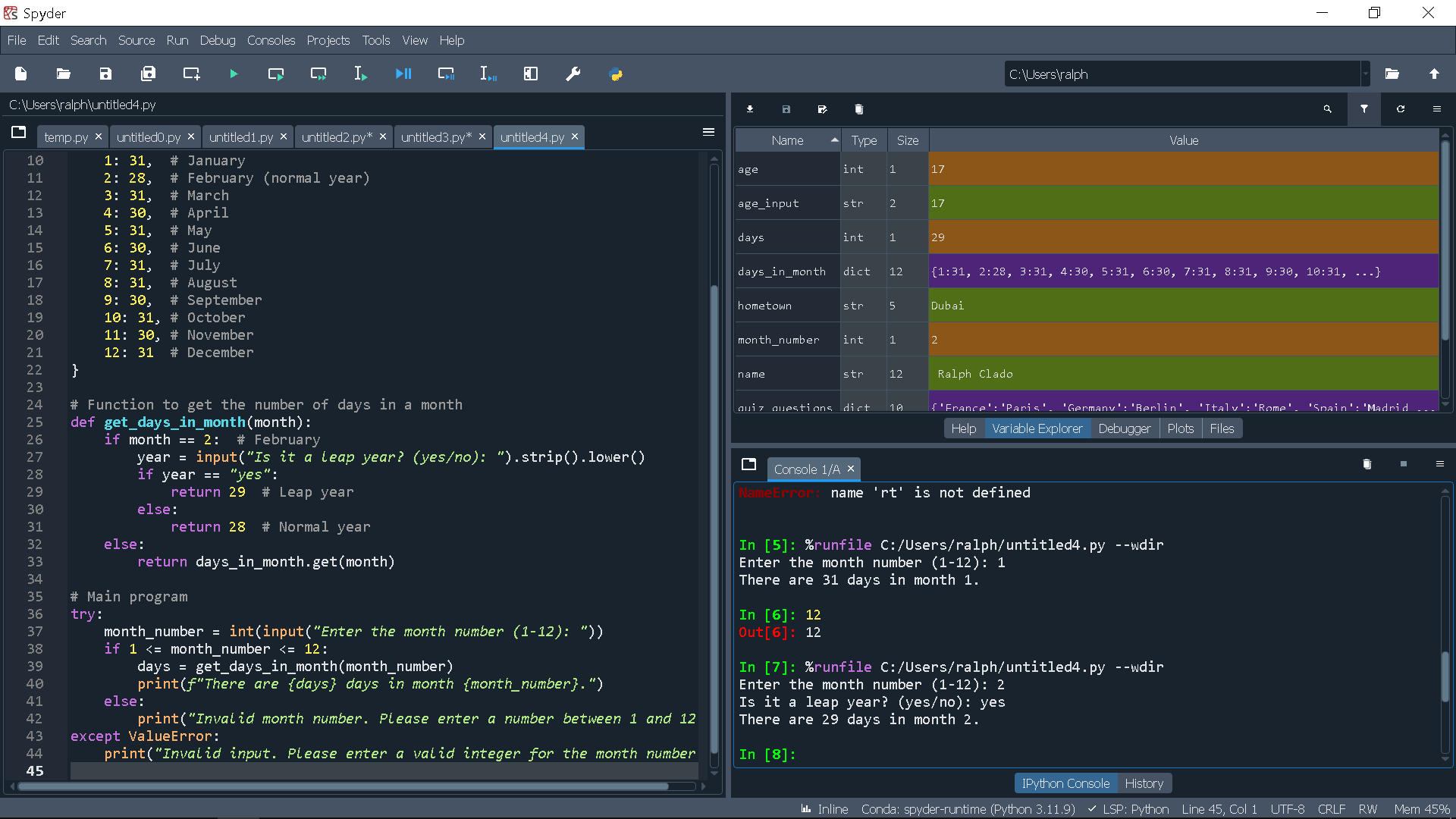
Dictionary: days\_in\_month keeps the number of days of each month, while February is set to 28 this can be altered.

Function: get\_days\_in\_month asks whether the month is February and, if it is, it asks the user if the year is a leap year, adding or subtracting the days correspondingly.

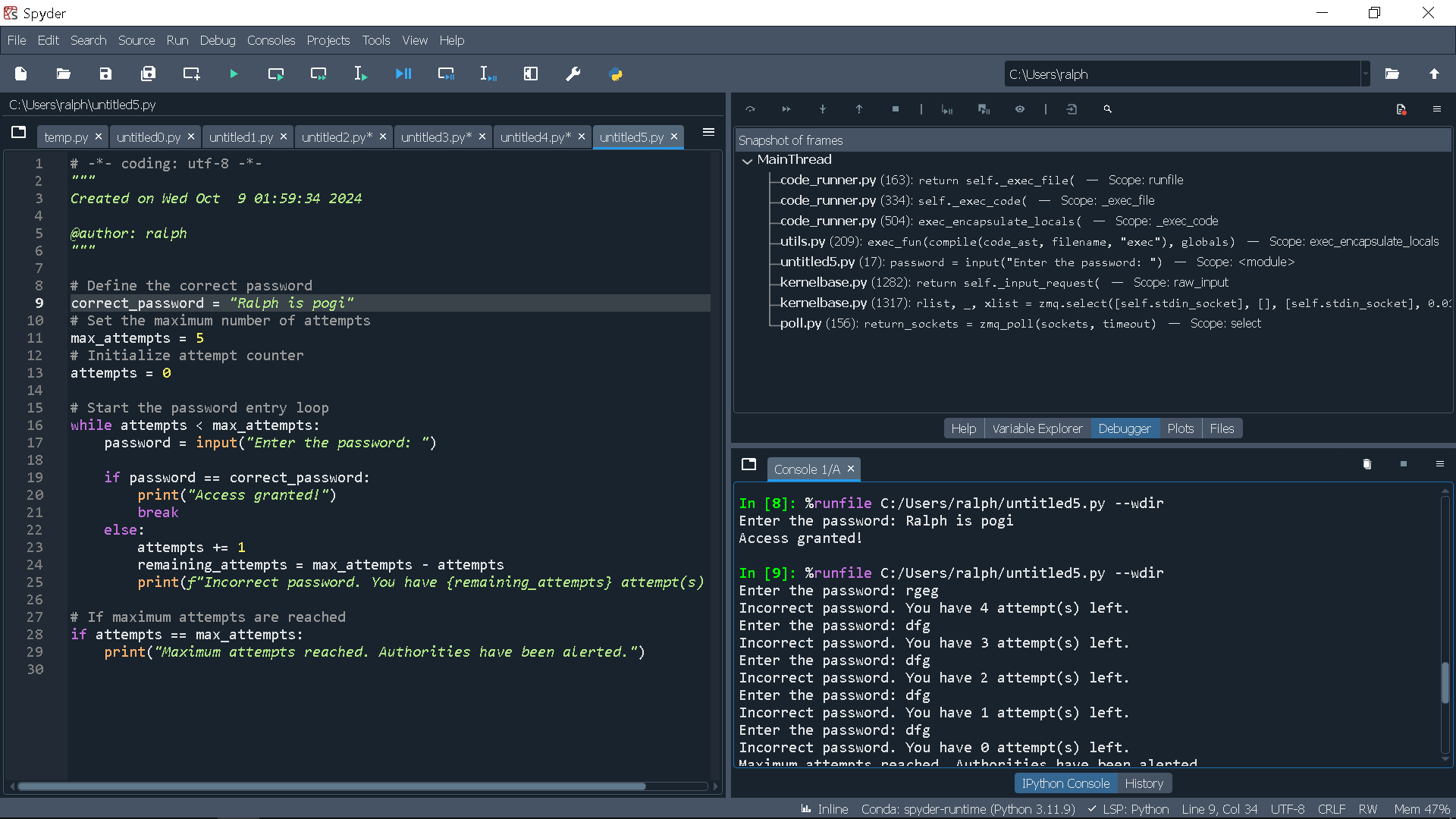
Input Handling: The program prompts the user for a month number and converts it to a valid month. Another is that it does a good job of clearing non-integer inputs through a try-except statement.

Output: Instead, if the input is invalid it gives an error message and if it is valid it prints out the number of days in the given month.

Exercise 5:



Exercise 6:



Password Definition: The right passwords are stored in the variable correct\_password.

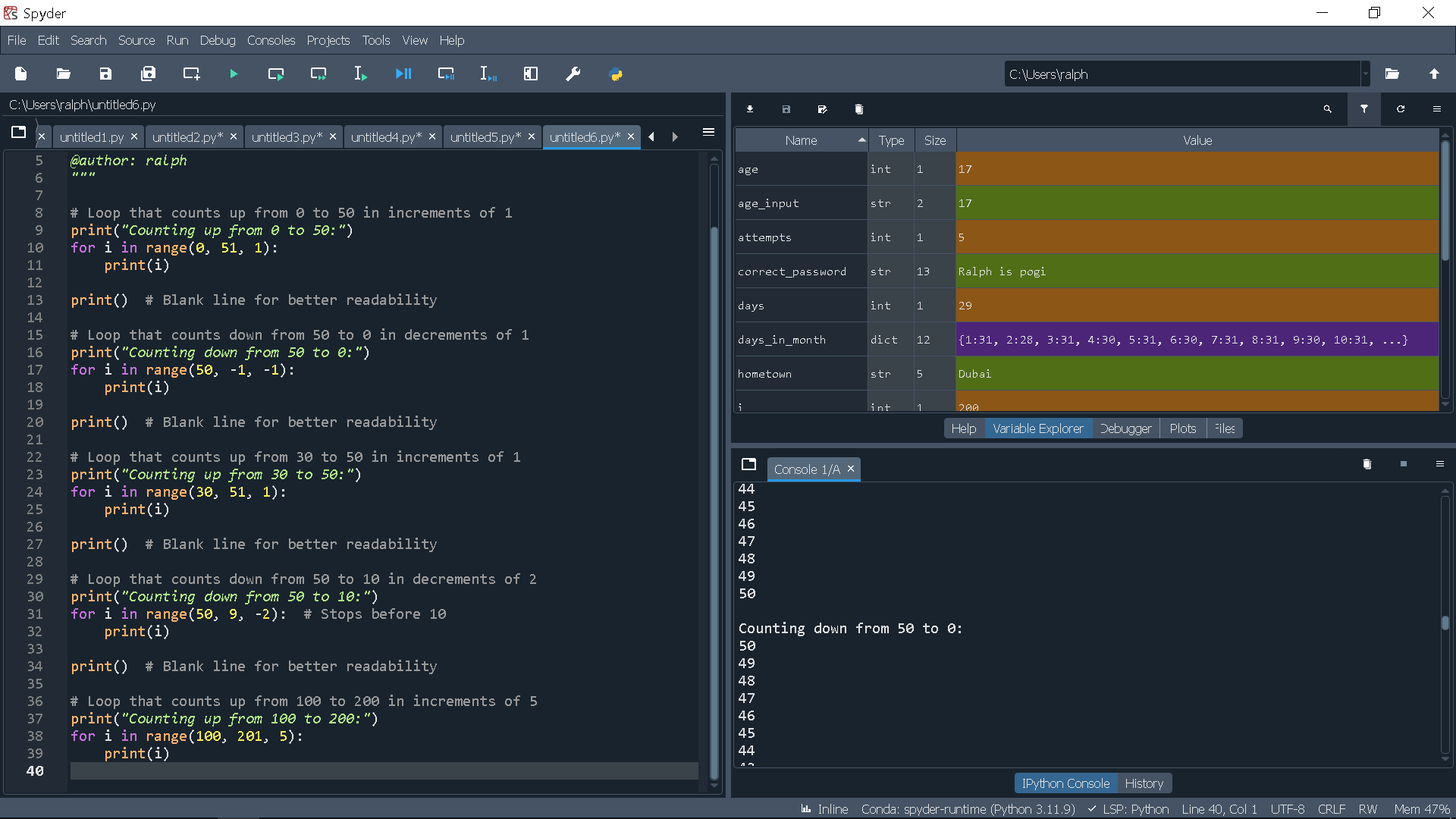
Attempt Limit: The program provides up to 5 attempts and is controlled by the max\_attempts counter and variable.

While Loop: This loop will keep asking the user for the password until he enters the right one or until the maximum attempts are typed.

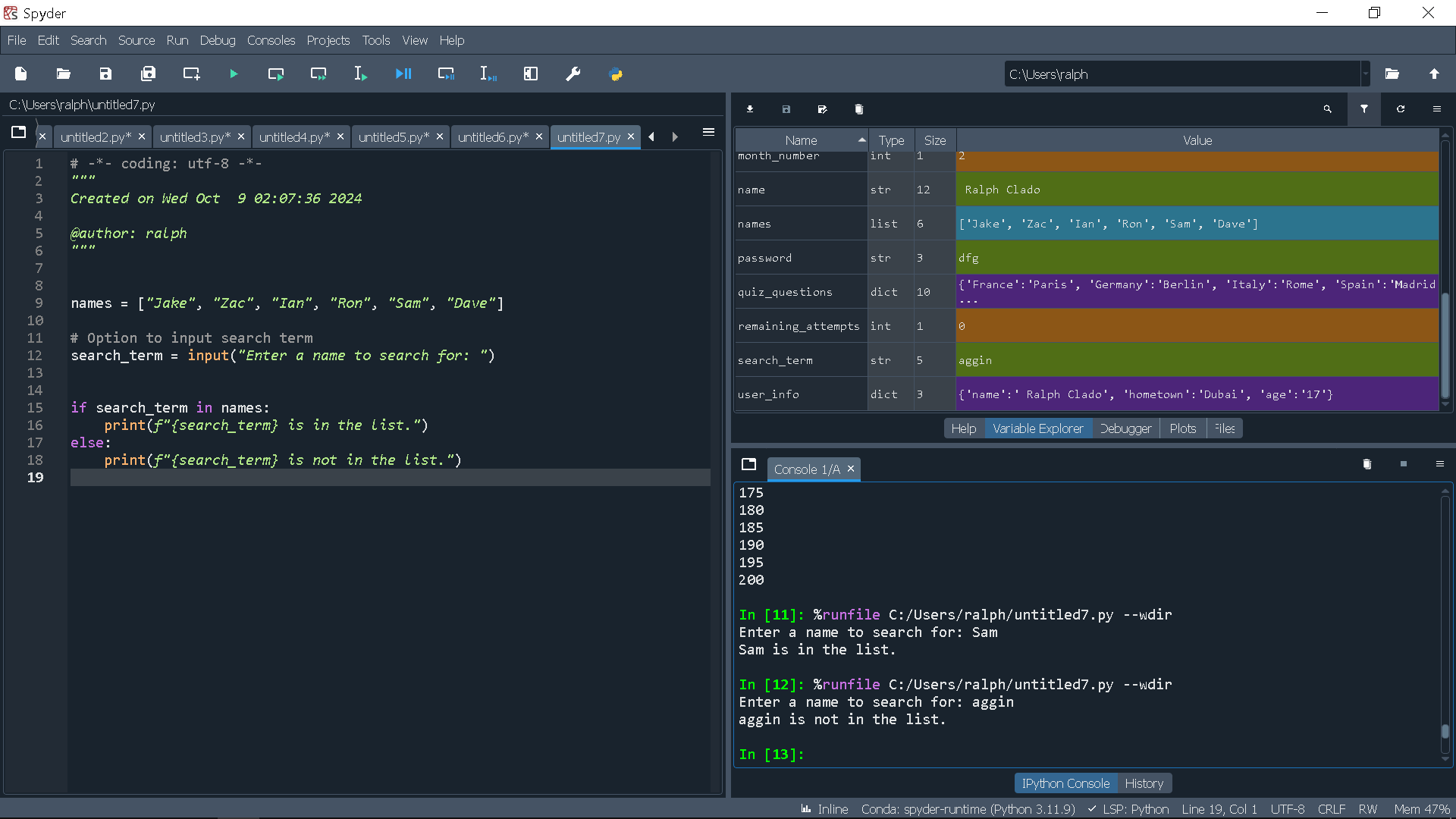
Feedback: Whenever a guess is made in error, the user is told how many attempts are left for guessing the password.

Final Message: In the event the user tries all possible ways, information is then passed to the authorities that the user has tried all options.

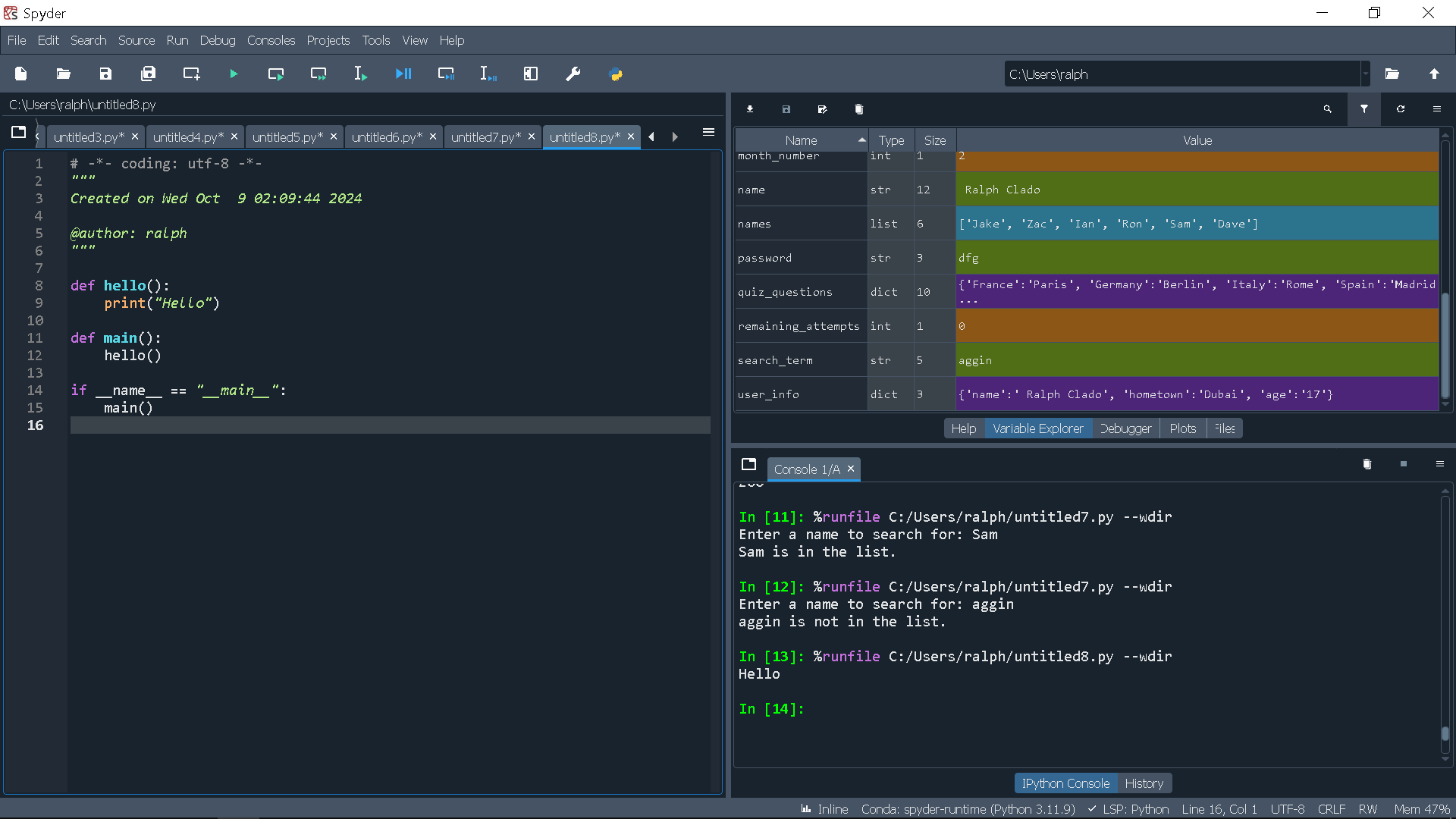
Exercise 7:



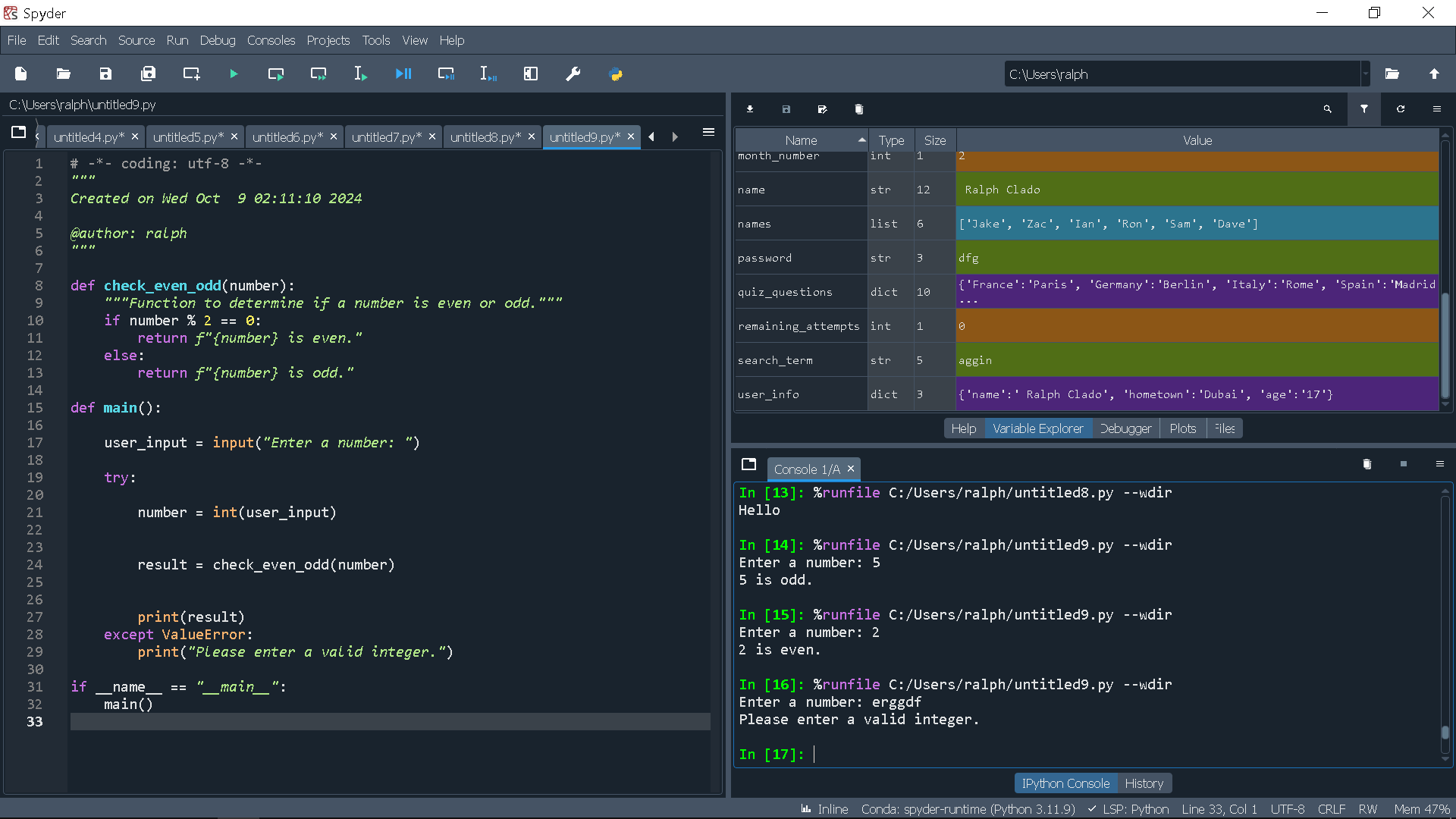
Exercise 8:



Exercise 9:



Exercise 10:



Function Definition: The check\_even\_odd function takes a number and then using the modulus operator (%) checks if the number is even or odd. Instead, it comes with a corresponding message.

User Input: The main function takes an input from the user in the form of a number and then type casts it into an integer.

Error Handling: WHENEVER the input entered is not an integer, the try-except block guarantees its validity. If not, it informs the user.

Function Call: The program then passes through the check\_even\_odd function, with the user number, and prints the returned message.