ECPI University Rafat Khandaker

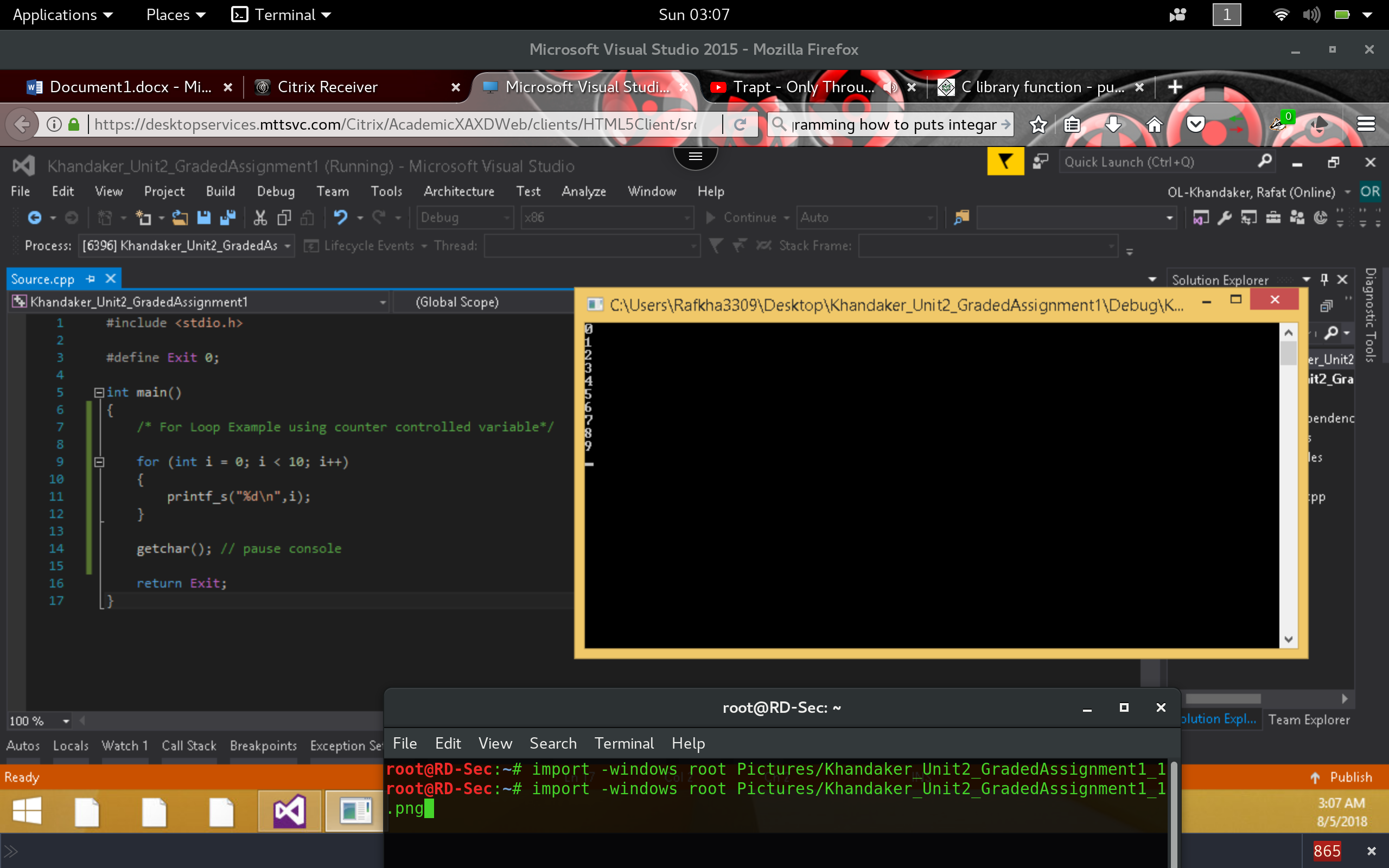
CIS\_126 08/04/18

Unit 2 Graded Assignment 1

Part 1

1. Describe how a counter controlled iteration structure works. Include each syntax item that is required to execute this structure successfully. Provide a C for() statement syntax example with your response.

*Counter controlled iteration ( loop counter ) is used to define the number of times a loop will execute. It is initialized with a pre-defined index with a value that represents the number of times the loop should execute. The variable is compared to a max or min counter & is usually incremented or decremented until the condition is false, in which then it will exit the loop.*

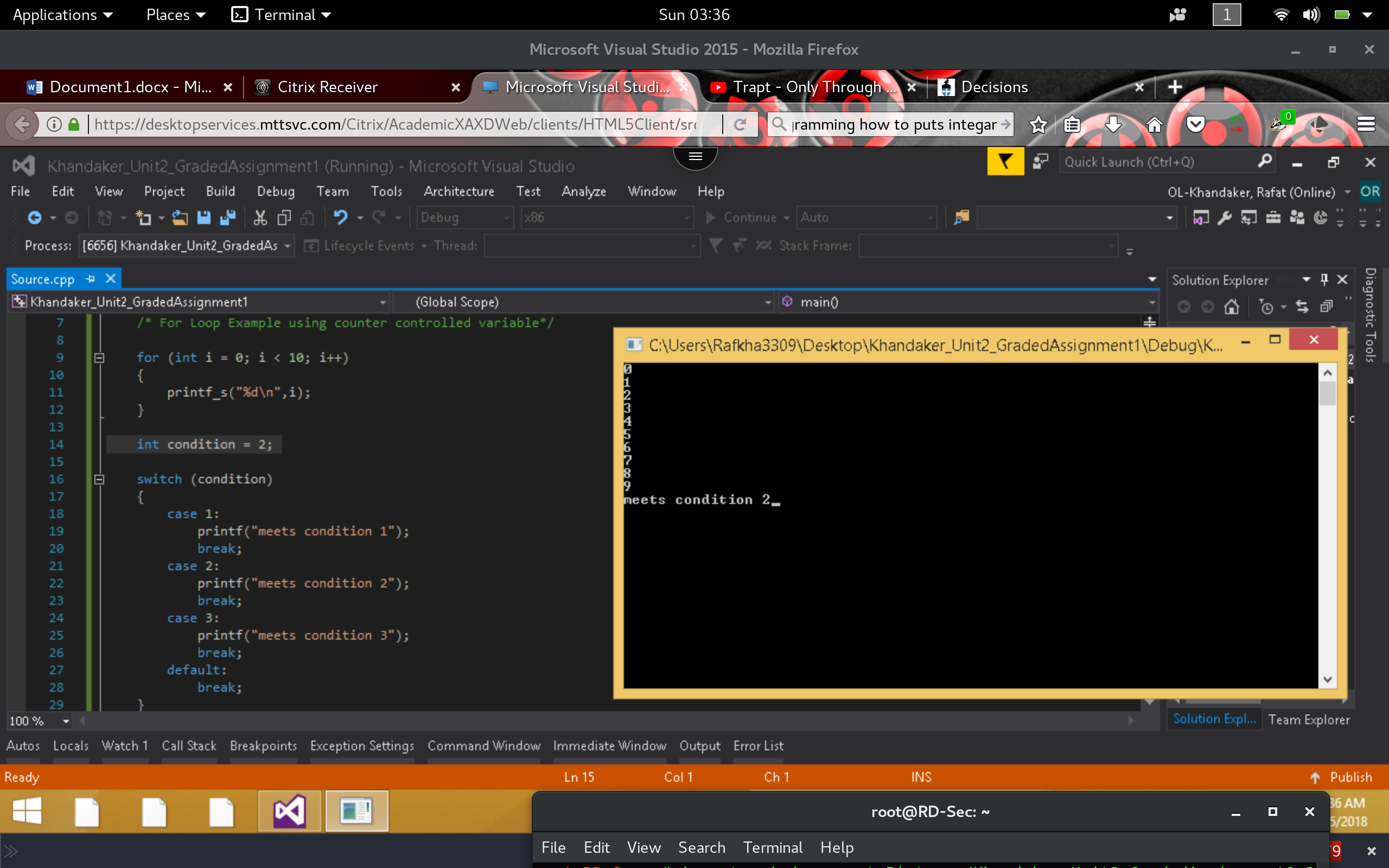


1. Explain the difference between a for() and a switch statement. Provide real world examples of when you would use each.

*For() statement is a looping control structure used to enter repeated task execution. Switch() statement is a decisional control structure, used to evaluate a parameter against a series of cases. If I were to loop through each value in a row of data within a SQL database. Similar to a cursor, I can use a for loop statement for this task. Example of switch case statement, I can capture a generic exception within an application & handle the details of the exception within a switch case statement,*

*logic ex: ( case exception is of this type? Then I will implement this task ).*

*In this example a for loop is used to iterate through a counter until I < 10. A switch statement was used to check if the condition matches the listed cases.*



1. Explain why we would use a function in a C program. What are the three items that are needed to implement this structure correctly?

*We use functions in C program to create modularity within our code, ability to break down code into smaller chunks. We use functions to allow Re-Usability of similar tasks without re-writing the program. In order to use a function, we need to (1) create a prototype of the function & declare it within our program. (2) Define the structure of the function; its return type and value or reference inputs. We need to create the function, somewhere in our program & develop the details of the function code. We need to pass in any arguments/parameters by either reference or by value also return value of the function. (3) Finally, we need to implement this function in our main function stack.*

1. With functions we can pass arguments by value or by reference. Please describe the difference.

*A function argument, passed in by value, will use the value of the function to implement its function task but will not alter the memory of the original variable. An argument, passed in by reference, will also alter the value in memory of the data object passed into the function if any changes occur.*

1. What is a recursive function? How does recursion relate to iteration?

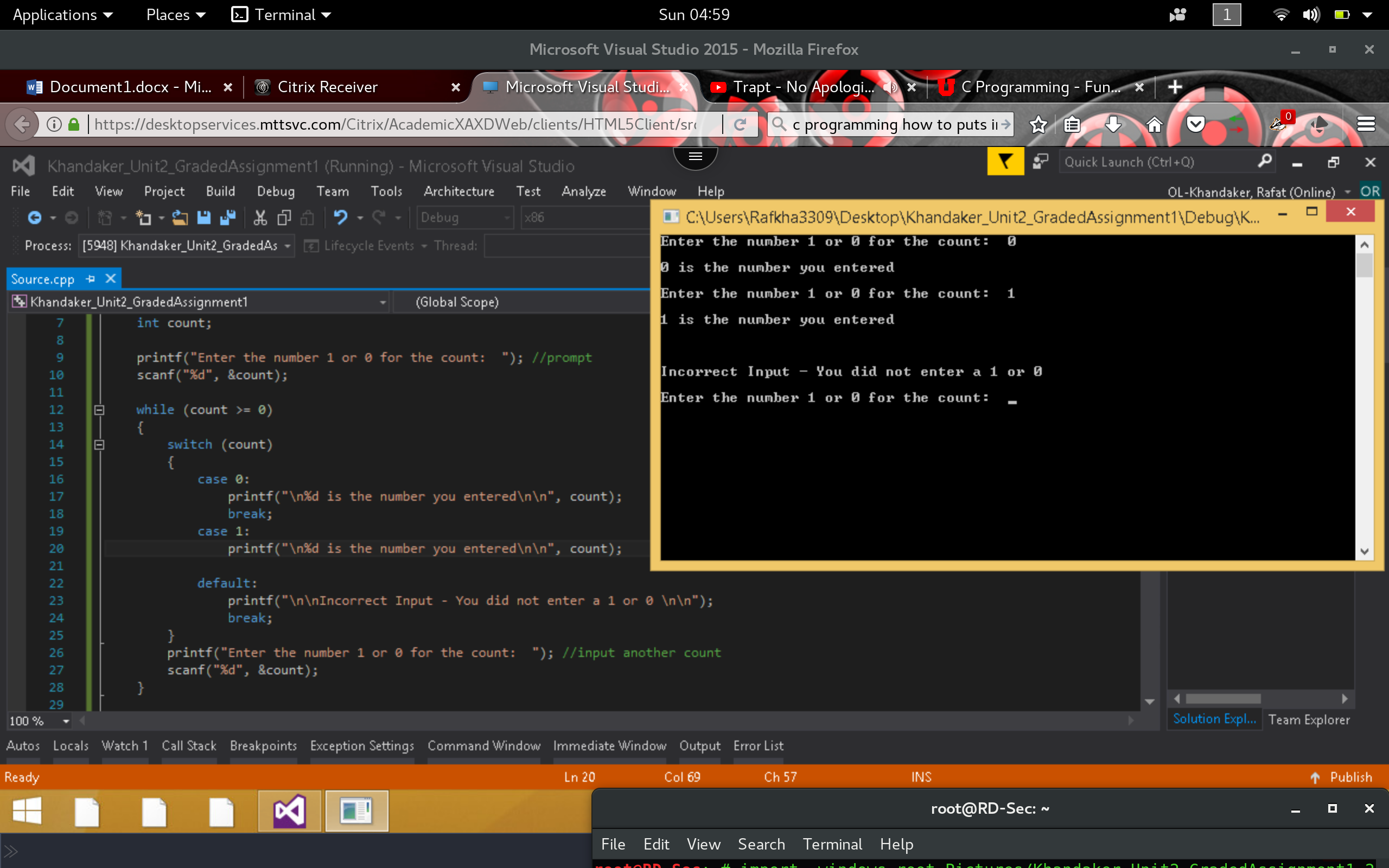
*A recursive function is a function that; in some way, will call another instantiation of itself, inside its own function code. By doing this, recursion is similar to iteration because it will repeat its own tasks until a condition inside Its code will break It's call stack. An iteration inside a loop is similar can be written as a recursive function.*

*Part 2*

1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code.

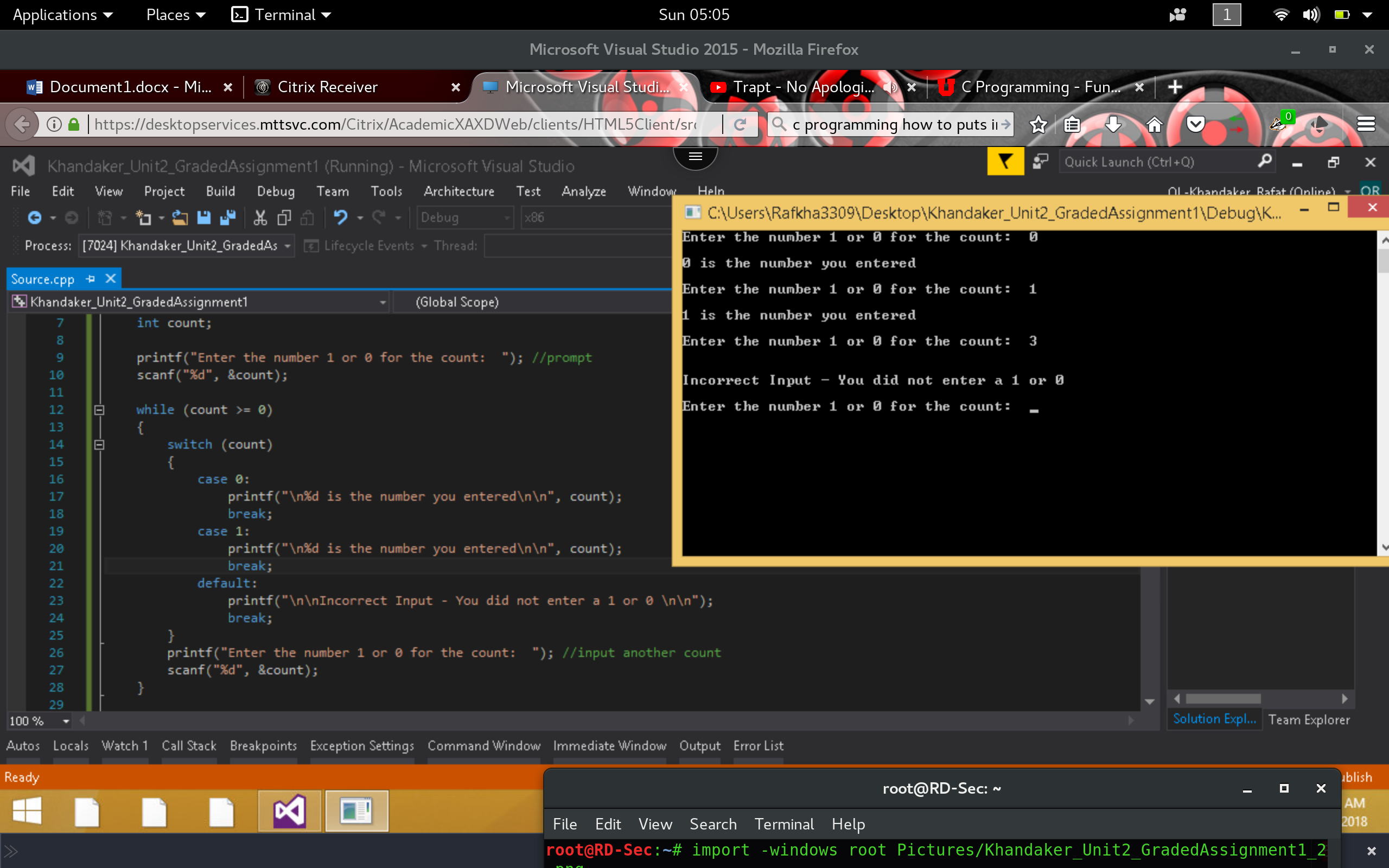
**incorrect**

*Running the program, I notice that when a 1 value is entered, we also get the improper message for invalid input.*



**corrected**

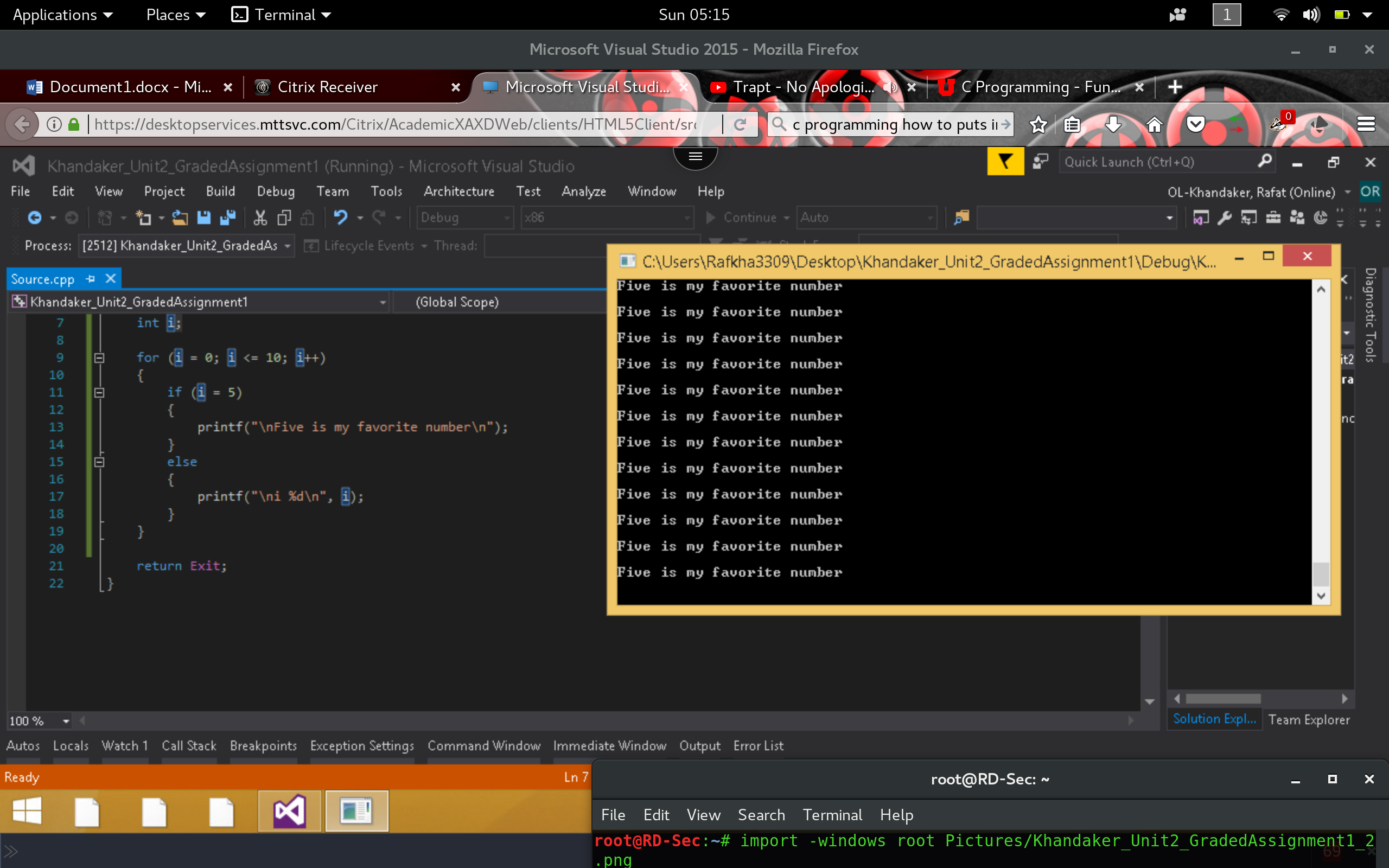
*By inserting a “break” in the case 1 statement, we will have the proper output for input a 1 value.*



1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code. Note: Be sure to check the output screen to see if the correct values are displaying according to the output statements.

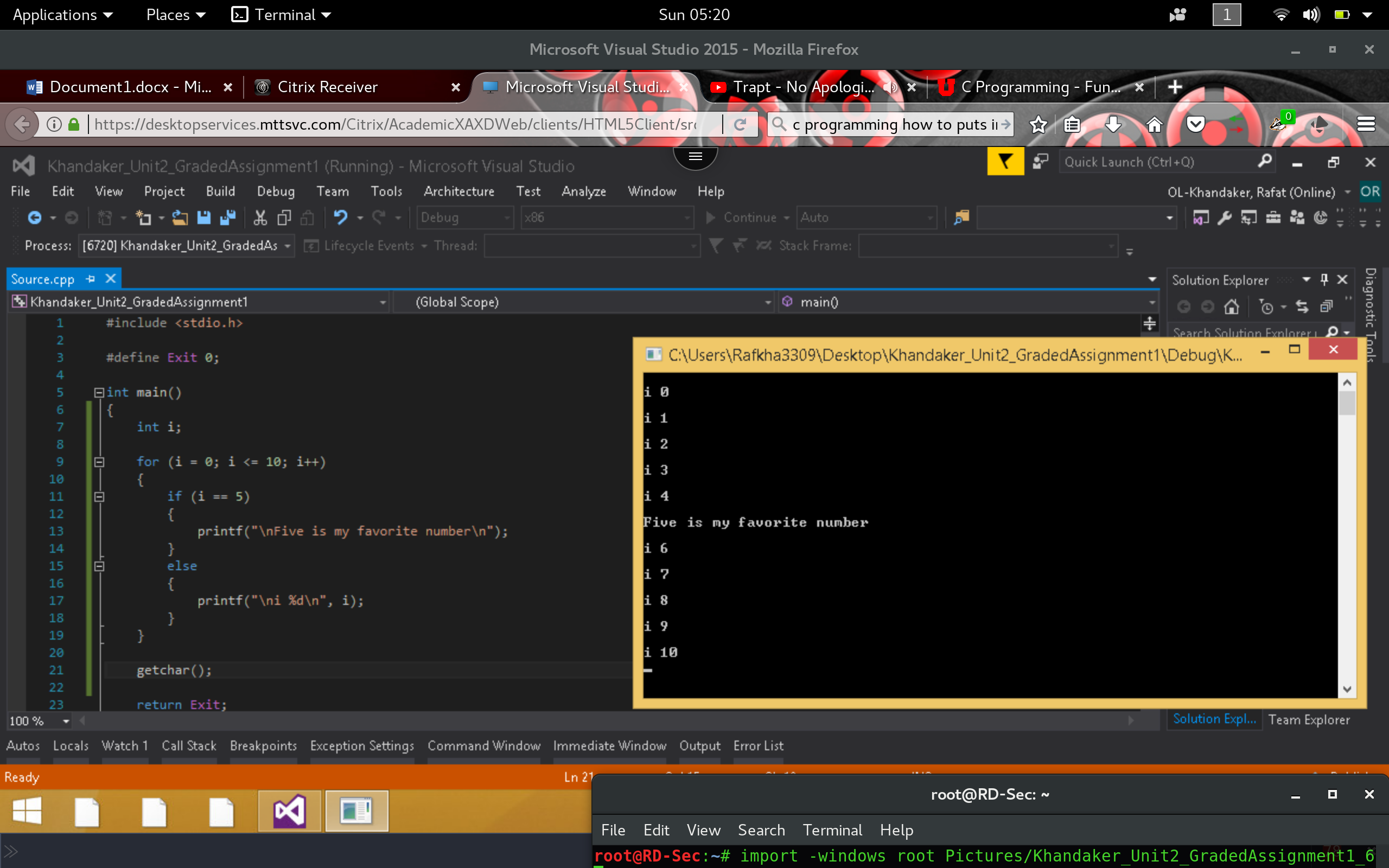
**Incorrect**

*Running the program, I noticed that the build was successful, yet my console seems to be stuck on an infinite loop.*



**Corrected**

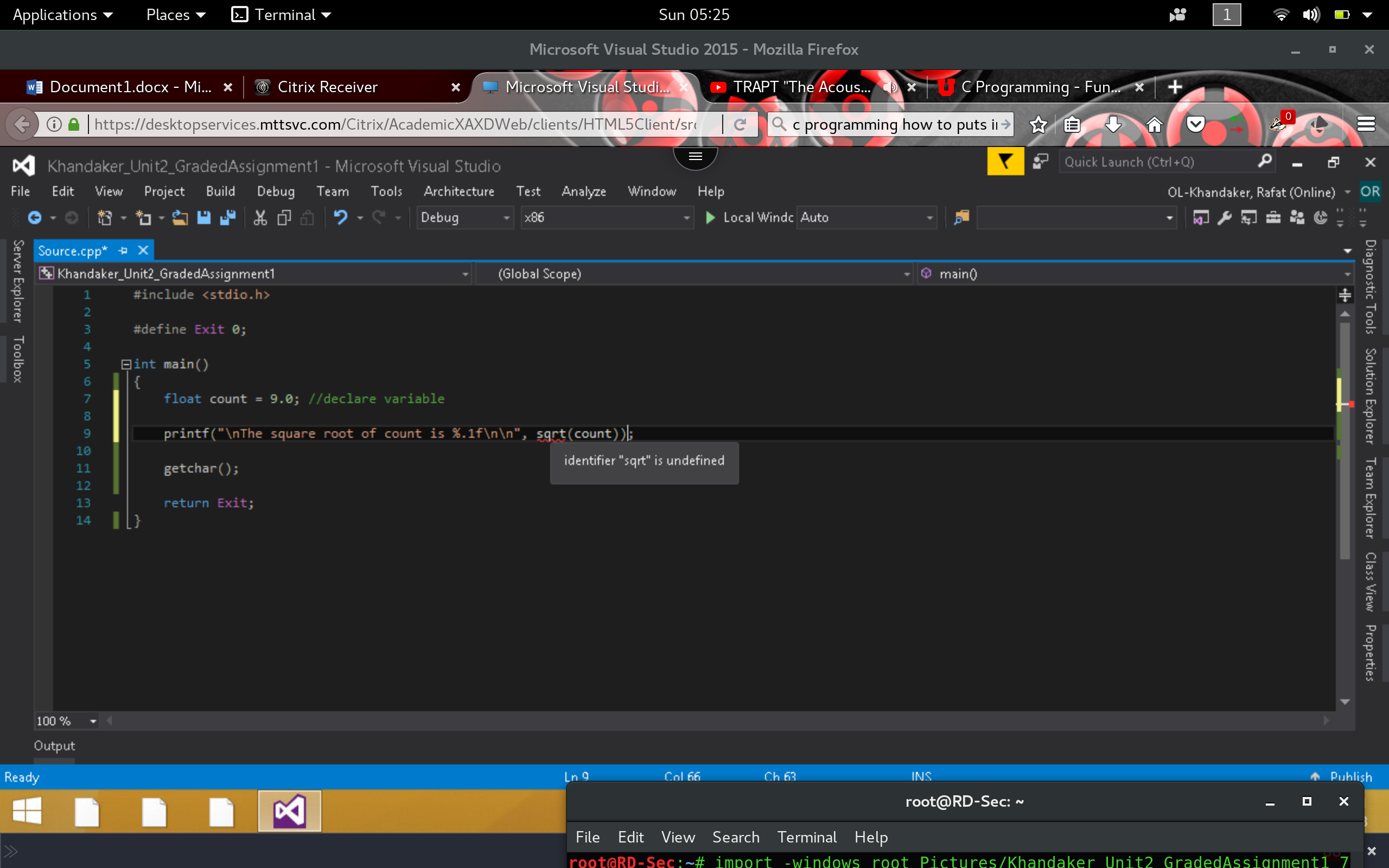
*It seems that in our “if” statement, we have assigned our counter value to 5 then print a statement, our loop is stuck infinitely between iteration 5 / 6. The “i = 5” should be “i == 5.”*



1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code.

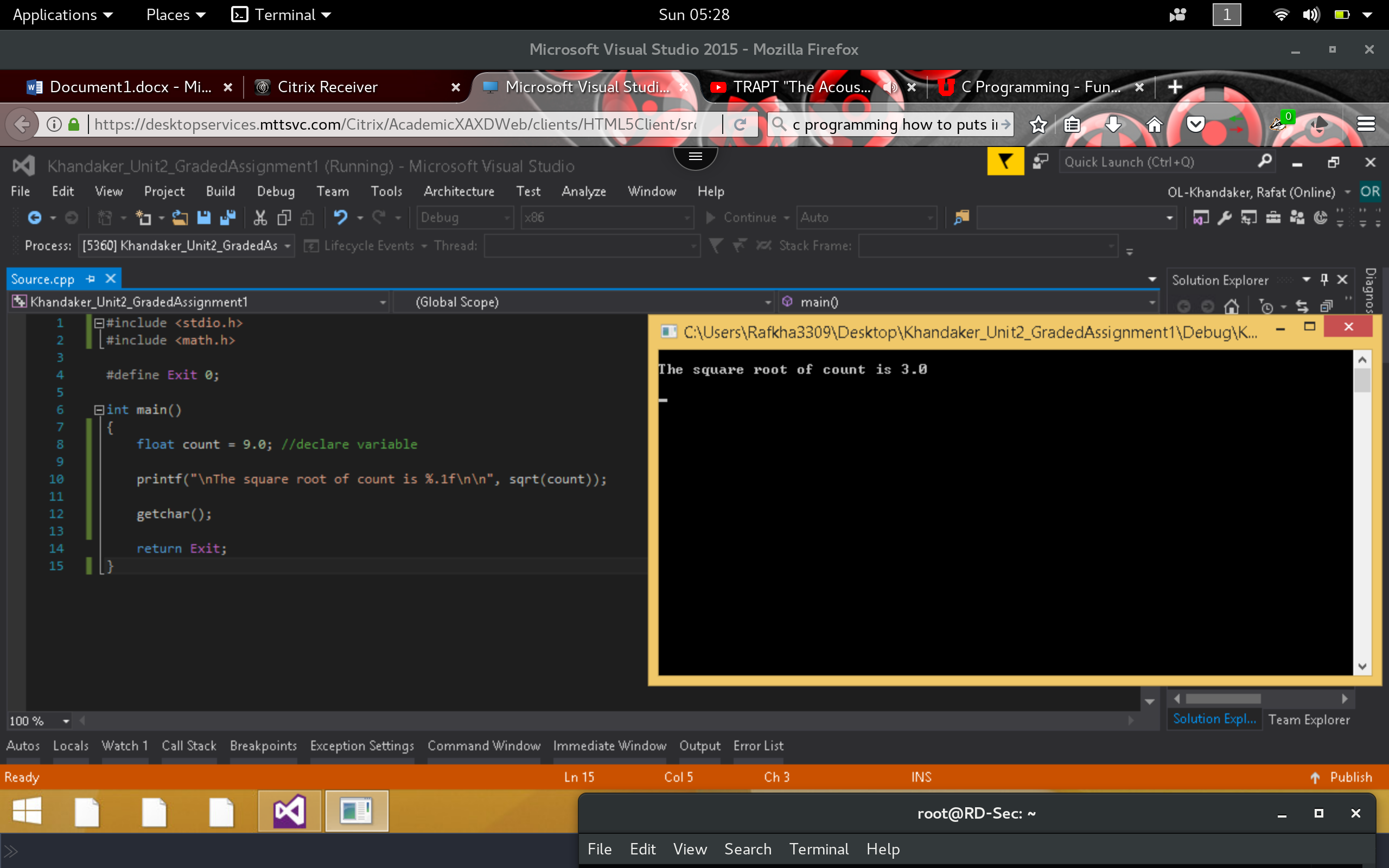
**Incorrect**

*Immediately our compiler shows us a syntax error, near our sqrt() function. This is most likely because the function is not mentioned or referenced.*



**Corrected**

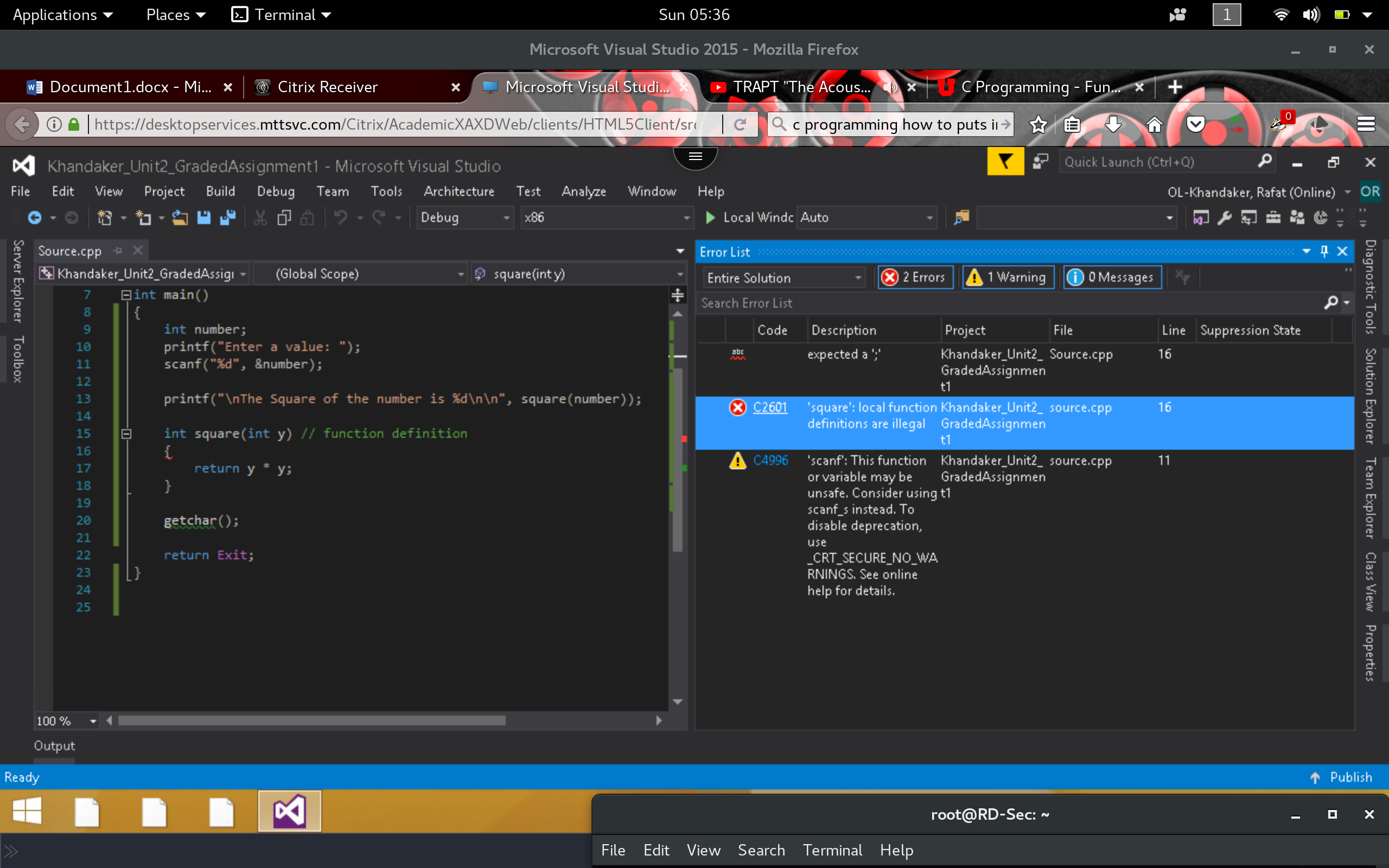
*By importing the <Math.h> header, we have successfully removed the syntax /compiler error. We successfully can run our program & display the proper output.*



1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code.

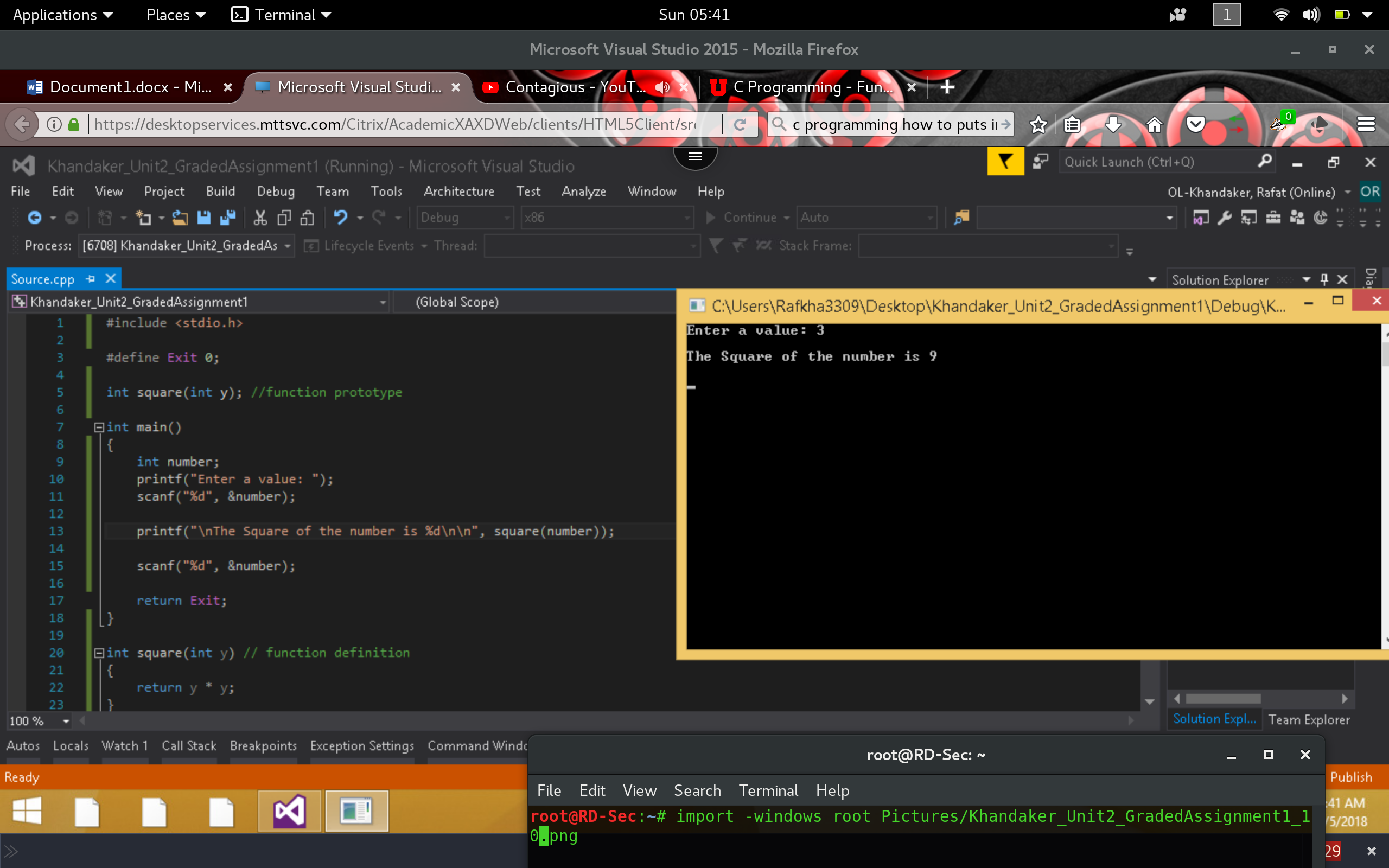
**Incorrect**

*In this example, it seems that the function was defined inside the main function stack, it should Instead be defined outside the main function.*



**Corrected**

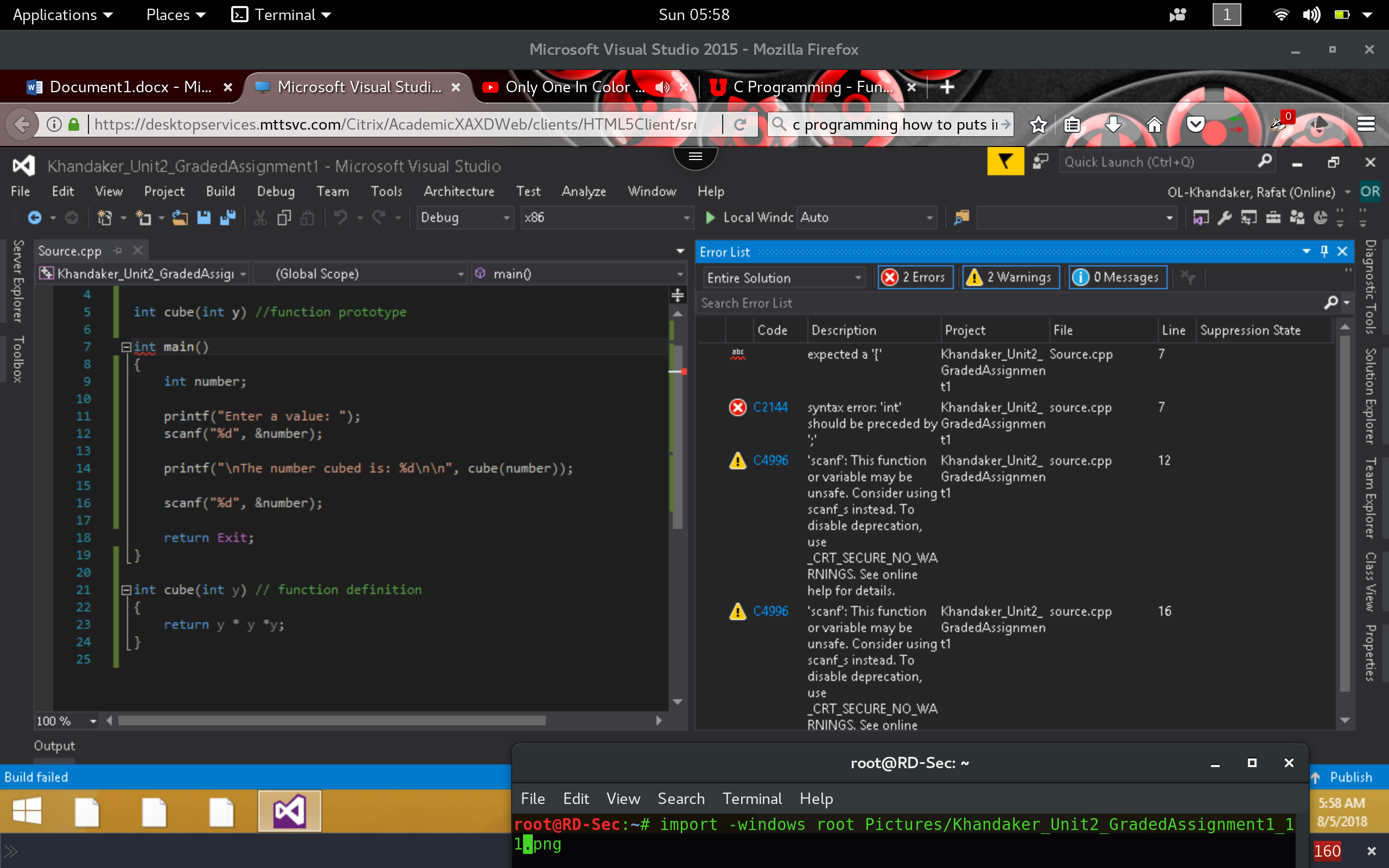
*By moving the function outside the main function stack, we will get the proper execution of this function.*



1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code. Note: There are two errors in the snippet below.

**Incorrect**

*In the example below, we have caught 2 errors in the compiler. Both errors can be corrected by inserting a semi colon on the function declaration.*



**Correct**

*Inserting the semi colon on the declaration will correct both errors and compile the program successfully.*

