ECPI University Rafat Khandaker

CIS\_126 08/19/18

**Unit 4 Graded Assignment 1**

**Part 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Describe how the following Input/Output Functions work: fgets(), putchar(), getchar(), and puts().

**Fgets()** *: is a function that will read a stream of string input & store the location in a pointer. We can specify the size of character we want to read and store from the stream input.*

**Putchar()** : *prints a single character to output stream.*

**Getchar()** : reads an input of a character & stores value as an int.

**Puts()** : prints *a string of characters and breaks a new line.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Chapter 8 in your textbook discusses different functions that can manipulate and search strings of data. Please list and explain one. Include the header file that is coincides with and provide a real-world example of when you would use it.

*There are many header files that can manipulate search strings contained in <string.h> , <ctype.h>or <stdlib.h> . String.h library for example contains, the following:*

**String.h**

* strcat - *concatenate two* ***strings****.*
* strchr - *string scanning operation.*
* strcmp - *compare two* ***strings****.*
* strcpy - *copy a* ***string****.*
* strlen - *get* ***string*** *length.*
* strncat - *concatenate one* ***string*** *with part of another.*
* strncmp - *compare parts of two* ***strings****.*

*A popular way to manipulating strings is by concatenating two strings with strcat. StrCat takes in two Char pointers in memory & returns a char pointer. We can combine two strings into 1 & store it into a single point in memory.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. In regards to output, what is a format control string? Please define its characteristics: conversion specifiers, flags, field widths, precisions, and literal characters. Provide a syntax example.

*Format control string is a what we use to display a modified output stream of a specific data type. Conversion specifiers is how we choose to format a particular data-type, ‘%d’ to print an int place holder. Printing precision is specifying the size of the value type to a degree, I can specify if a float value should be printed up to 2 decimal places with ‘%.2lf‘ or print/scan up to 20 characters with ‘%20s’ .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. When reading formatted input using a scanf() statement we have to make sure that we include the correct conversion specifier which needs to match up with the variables declared datatype. Please name the data types that correspond with the following conversion specifiers: %d, %c, %s, %f, and %p.

*‘%d’ is assigned to an int variable type. ‘%c’ is assigned to a char variable type. ‘%s’ is assigned to a string variable type. ‘%f’ is assigned to a floating variable type. ‘%p’ is assigned to a pointer or memory address format.*

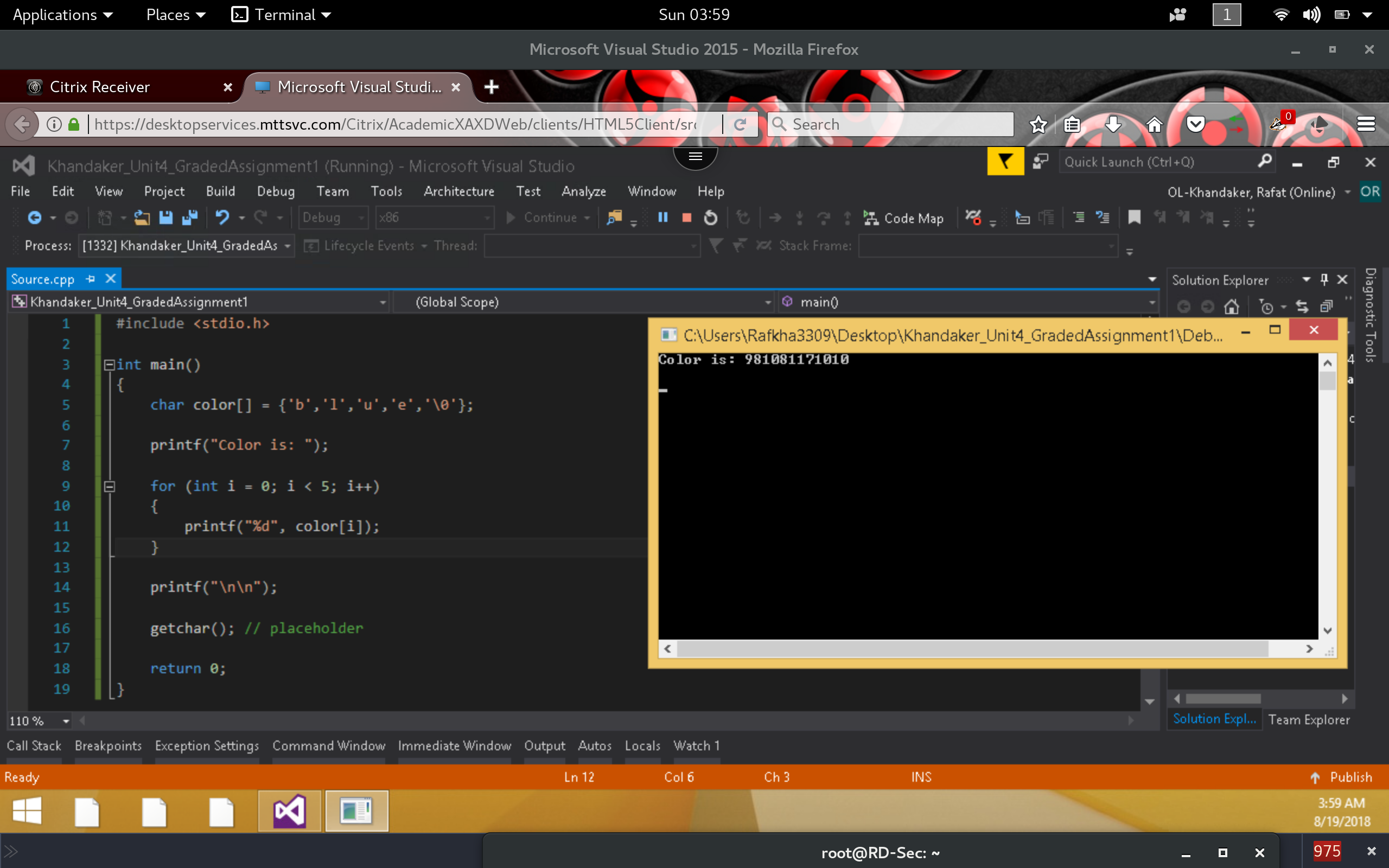
*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**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.

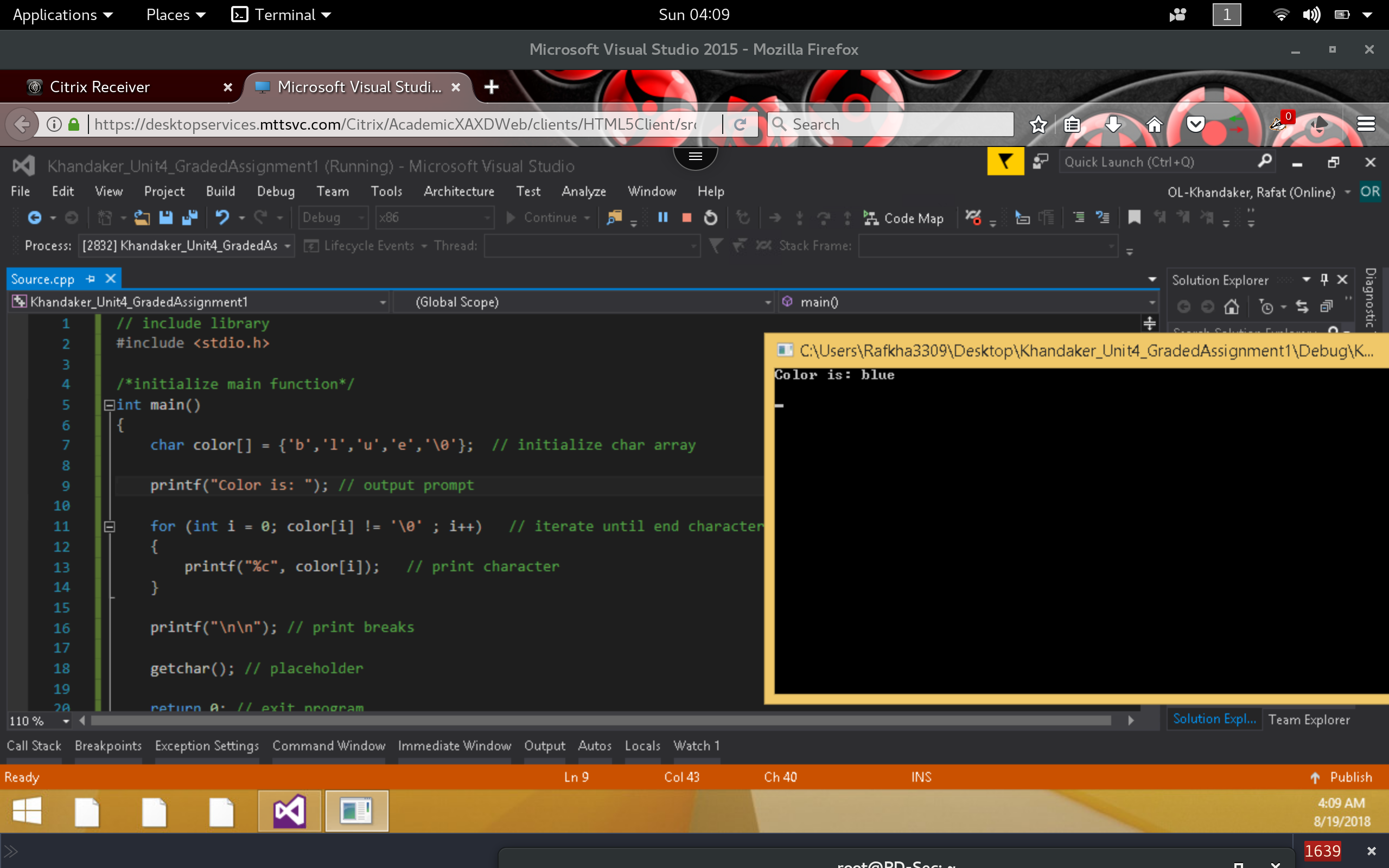
**Compiled Code:**

*In this program, we obtain the wrong values for color[]. We obtain these values when we attempted to print the output using the incorrect character conversion format. We also iterate passed the bounds of our char array.*



**Corrected version:**

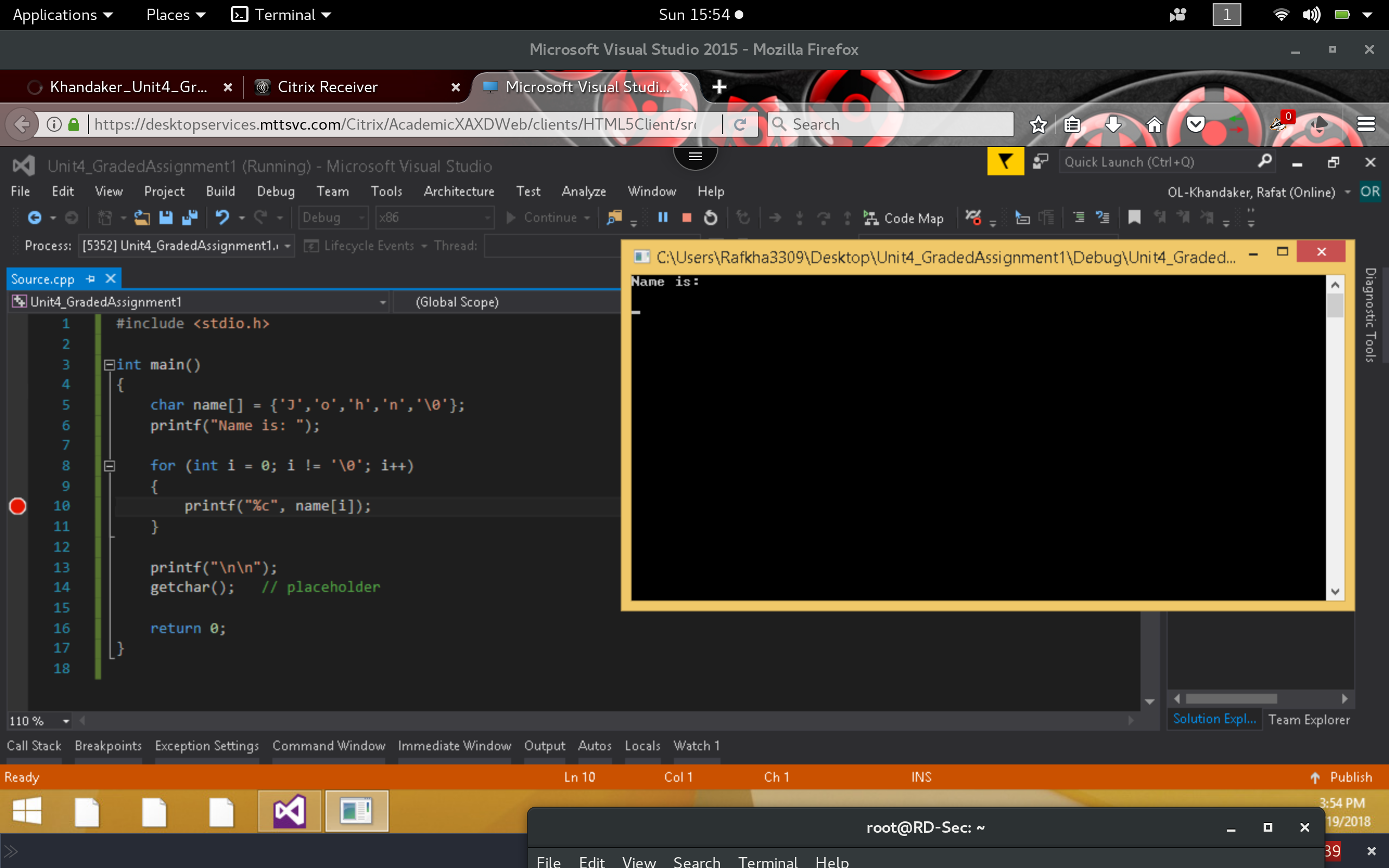
*To correct this issue, I will use the correct string conversion specifier. ‘%c’ as the output of a character. I will also make sure the loop does not iterate passed the string by stopping at the null pointer value ‘\0’.*



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.

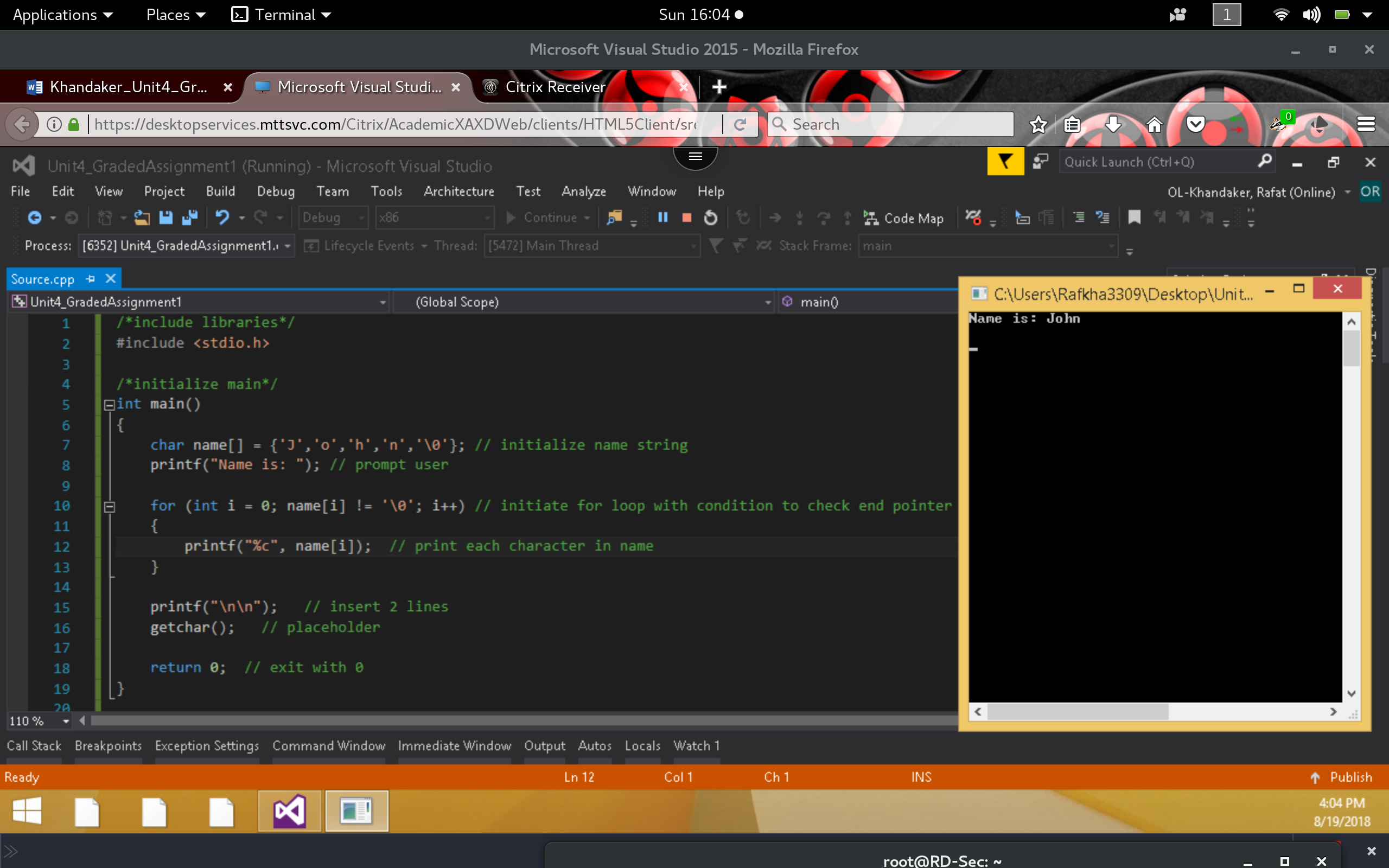
**Incorrect Code:**

*In this example, I placed a breakpoint in the for loop to verify if the print statement to print name will execute. Seems that ( int I = 0 != ‘\0’ ) evaluates to false & does not execute the for loop. This statement should be compared to name [] at index ‘i’ & not ‘i’ itself.*



**Corrected Version:**

*To correct the incorrect output, I will replace condition of for loop to check the index of the name string. We obtain the correct output.*



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

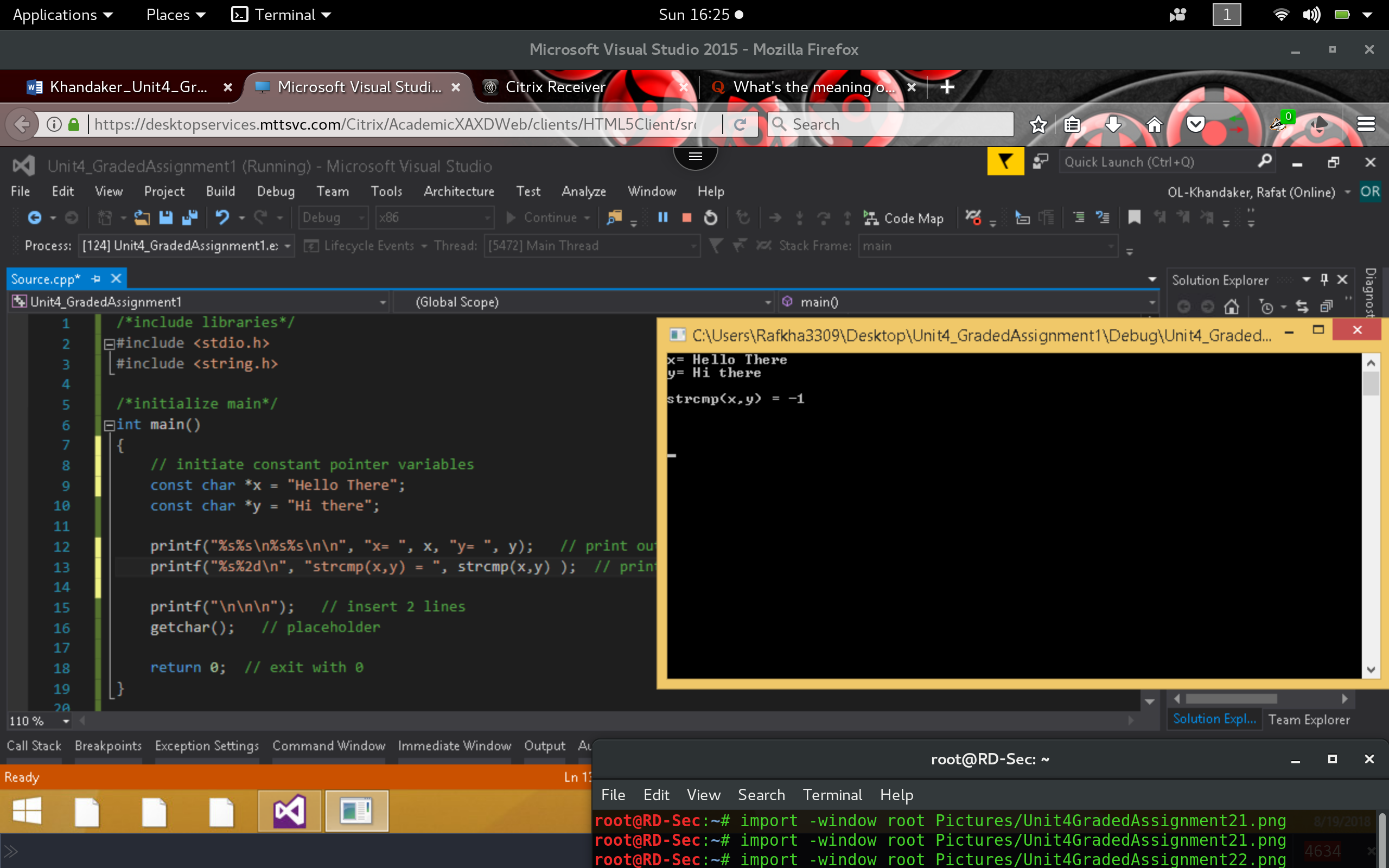
**Incorrect:**

*In this example, we have received a compile time error. The 3rd printf statement accepts too many arguments.*



**Correct*:***

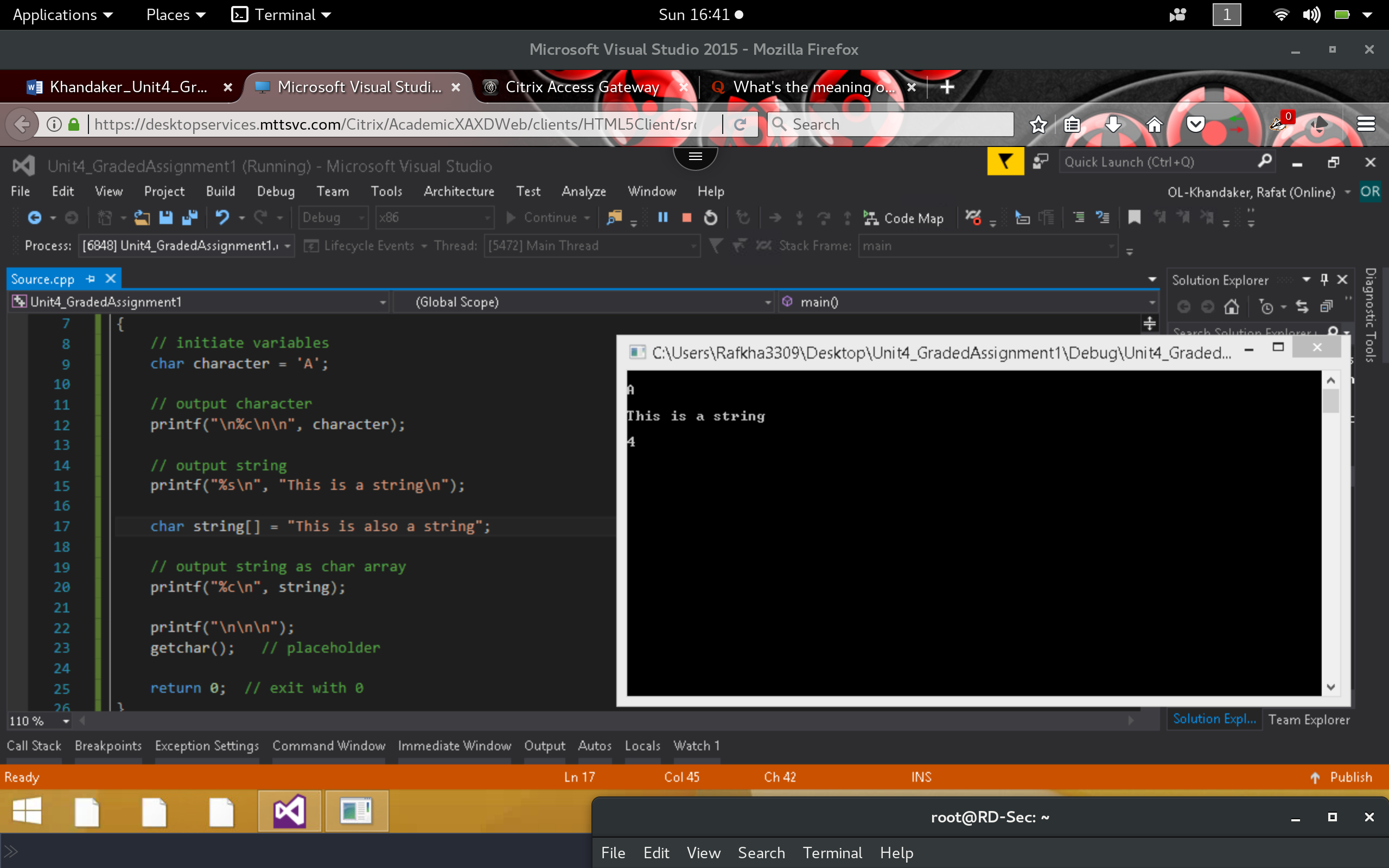
*By Deleting the 3rd printf statement, with incorrect parameter, we obtain the correct ouptut*



1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code. Be sure the output looks exactly like the screen shot below.

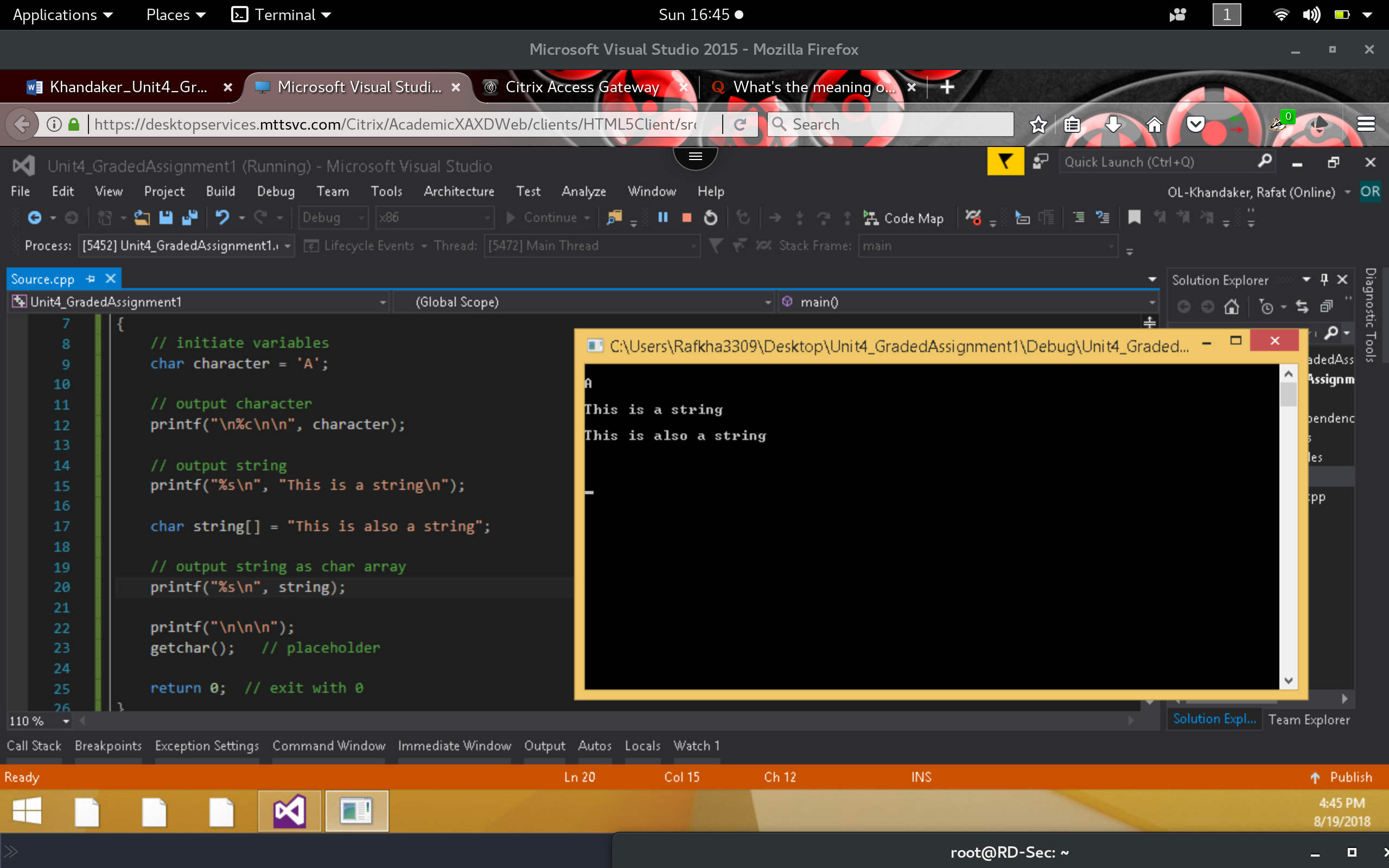
**Incorrect:**

*In here we receive no compile or runtime error but our conversion string specifier for the last printf statement is incorrect. We need to print the value as a string & not a character*



**Correct:**

*In here we simply specify the conversion string as a string ‘%s’ & not a character ‘%c’ this will display the correct output.*

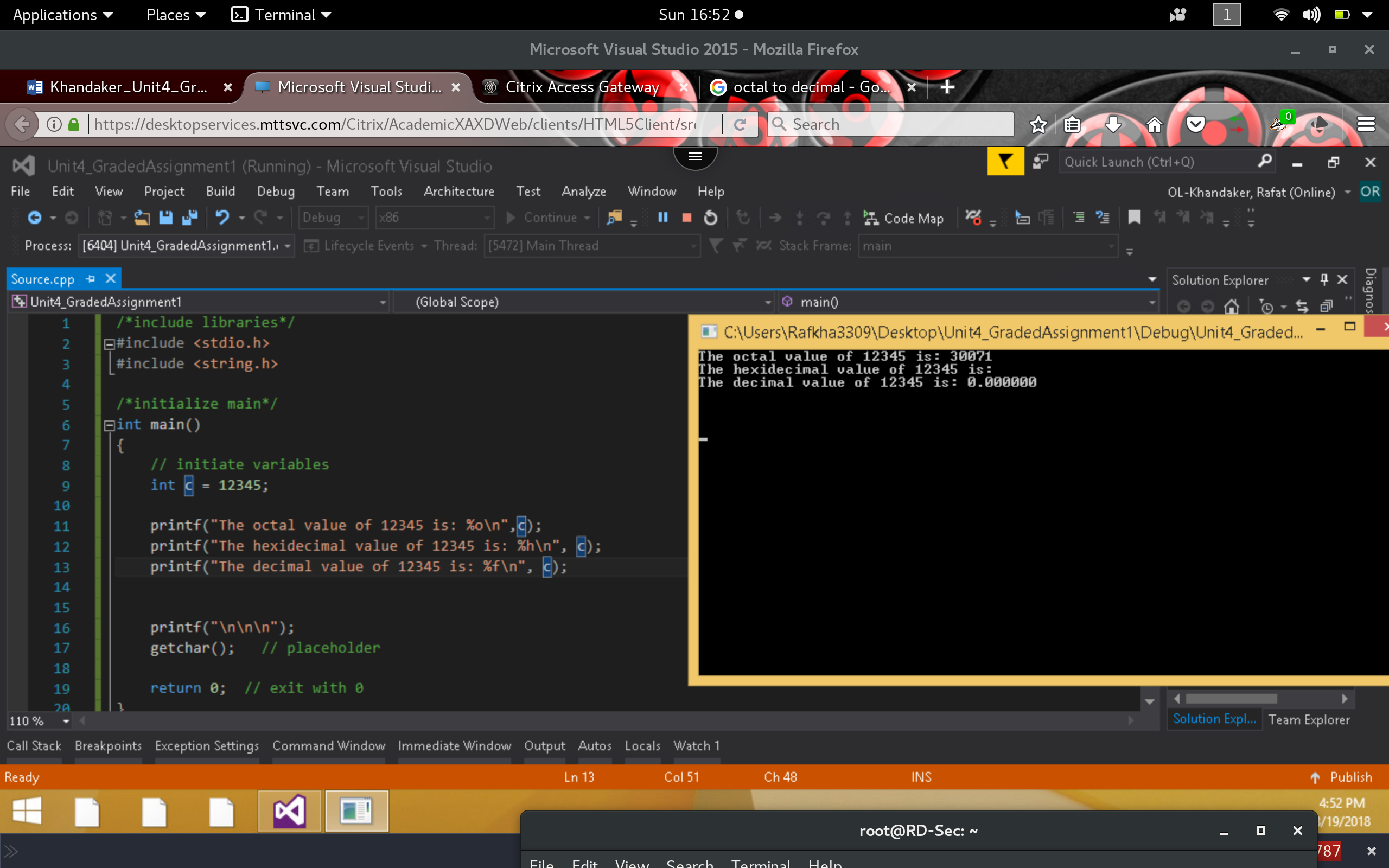


1. Execute the following code and identify the errors in the program. Debug the program and provide the correct version of the code. Adjust the code to look like the output in the screen shot.

**Incorrect:**

*In here, we have the wrong conversions output of integer values to other conversion types.*

*Values for last 2 print statements are incorrect.*



**Correct:**

*By changing the conversion string to ‘%x’ we print the correct hex value.*

*By chainging the conversion string to ‘%f’ & casting the int as (float) we obtain decimal*

