ISSN: 2167-1435
v31 n3005

Decomposition and Reconstruction of Complex Spreadsheet Functions

William J Tastle
tastle@ithaca.edu

Ute St. Clair
ustclair@ithaca.edu

Loden Cullar-Ledford
lcullar1@ithaca.edu

Department of Management, School of Business,
Ithaca College
Ithaca, New York 14850, USA

Abstract

This paper addresses the difficulties in teaching students the creative thinking skills necessary to assemble some straightforward string manipulation functions, by themselves limited in scope, to produce new output that can be analyzed using traditional worksheet methods. The manipulation of strings is very useful in preparing data for use by either Excel or virtually any other program. In fact, Excel is an excellent tool for modifying data for importation into a database. The student is shown how to parse a block of raw data using string manipulation functions, such as the LEFT(), RIGHT(), MID(), FIND(), and SUBSTITUTE(). Critical to utilizing these functions effectively is knowing how they work and what mental models one must create to combine functions creatively to yield a proper solution. The process involved for each of these is described by means of a detailed illustration. The instructor will be able to use these tested techniques to provide students with increased mastery over the spreadsheet tool.

Keywords: Strings, string manipulation functions, nesting functions, decomposition, reconstruction.

1. INTRODUCTION

The spreadsheet is the *de facto* analytical tool of choice in the business world for one rather obvious reason: virtually every computer in business has a spreadsheet program, and it is usually Excel. The range of usage for the spreadsheet is vast, from its use as little more than an accountant's pad, onto which labels and numbers are typed and the numbers totaled, to a sophisticated modeling- and analysis tool used in data mining, business intelligence, and analytical-based prediction.

There is much to be gathered from previous work in teaching students how to think in a critical – out-of-the-box – manner using what tools they have at their disposal to find a solution. Finance students learn their formulas but, when given a highly unstructured problem, seem lost (Carrithers, Ling & Bean, 2012). If a spreadsheet is considered a "mindtool" (Jonassen, Carr & Yueh, 1998) in which

"learners use [spreadsheets] to represent what they know, [and] necessarily engage them in critical thinking about the content…" then they discover that the spreadsheet can be somewhat of an artist's palette ready for creative use. Certainly, the costs of not having developed the skill of thinking are pervasive (Caulkins, Morrison & Weidemann, 2007). Using a spreadsheet as a "forced" organization tool is certainly a useful exercise, and it does force some little bit of creativity from the student, but the full force of the spreadsheet as a mindtool by which students learn to think in structured ways, putting organization to disorganization, is missing. To force the development of creativity in the mind of the student requires a task that cannot be accomplished without that skill.

Somewhere in the middle of this vast assortment of spreadsheet usage is that of the nesting of functions into complex arrangements. In this paper a method is described that greatly facilitates the understanding of nested functions. Although this method can be used to master any combination of nesting, it was developed specifically to help students understand the complexities of string function manipulation; so it is in the spirit of innovation that this paper uses strings to explain the method. The authors has chosen to use several Excel functions to illustrate the method.

2. CONCEPT

Every Excel function contains an argument that might contain nothing, as in the case of =NOW(), or several elements, such as *=SUBSTITUTE(cell, old character(s), new character(s), which instance of the old character to replace with the new character)*. Sometimes it is necessary to use additional functions to evaluate a condition or calculate a value that is used within another function. A generalized function would be:

$$= var_1(arg_{1.1}, var_2(arg_{2.1}, var_3(< char >, < cell >)), \ldots, var_n(arg_{n.1}, arg_{n.2}, \ldots))\ldots)$$

where var_i is some spreadsheet function. 'Cell', 'range', 'char', and 'integer' follow their usual meanings. An unsophisticated user might calculate var_2 through var_n in separate cells, and then combine them in the final var_1 to arrive at an answer. Such a method would leave the spreadsheet possibly covered with temporary functions, spurious numbers, and an overall structure that is very difficult to manage and revise. Accidently deleting a cell might cause an entire worksheet to be full of #REF! errors, and one might be unable to reverse the accident. Therefore, the solution is to spend an extra few moments thinking through a solution that keeps all the functions needed for a single action in one formula.

3. FUNCTIONAL DECOMPOSITION AND RECONSTRUCTION BY LEVEL

Suppose that one needs to extract a substring from a cell and that the data are consistent. Consistency identification is a necessary skill a student must develop, in order to manipulate string data. Sometimes one can take advantage of the "text to columns" wizard in the Data tab (Excel 2013) to partition the data into separate columns, and that works when the data are either of fixed length or when there are some consistently placed characters used to separate the data components. For example, if a semicolon or a comma is used to separate each field, then the "text to columns" wizard can work quite well. All too frequently, unfortunately, legacy data or Internet data are not so graciously presented, so that the resulting data must be inspected for consistencies. If no consistencies are identified, one must insert one's own, in order to separate or parse the cell. It is especially in this latter case that nested functions and associated complexities arise that are a challenge for students without a programming background to master and for instructors to teach. The authors have identified three straightforward rules, along with illustrations, which show how this complexity can be greatly simplified.

A complex string function can be simplified by writing it in levels in which each succeeding nested function drops to a lower level. The rules follow along with an illustration of how they work.

RULES:
1) Functions that are only one "level" deep (i.e., functions lacking nested functions) are not decomposed to a lower level.
2) After every named function (i.e. IF, FIND, SUBSTITUTE, etc.) the contents of the argument are placed at a decomposed level beginning immediately after the left parenthesis. Only functions matching Rule 1 are exempt from this rule.

3) After the function argument is completed, the right parenthesis is raised to the level of the original function.

Suppose that someone has the following data set form which the student name, both the ID number and the Cell number as separate fields, and the email address must be extracted. The text-to-columns method (not discussed here because it is so basic to any level of worksheet competence) can easily separate the student name- and email fields using the comma as the delimiter, but the ID number and cell number will be placed in a single column, because there is no delimiter separating them. After guiding students through the text-to-columns data separation, such that they understand its limitations, it is desirable to revert back to the raw data to separate all of the fields using string manipulation. In other words, now that students understand that some of the data can be parsed using the text-to-columns method, the authors will shun the "built-in" approach in favor of learning how to use the text manipulation functions on all the data. The first thing to do is to identify the consistencies, for without consistencies in the data, there is little to be done. Fortunately, consistencies are typically found, once one has developed the insight; in those cases where none are apparent, one can creatively make one's own consistency.

The obvious consistencies are:

1. The presence of a comma following the student name field. In fact, both the comma and the subsequent space are consistent in all the data.
2. Following the string of digits is another comma and space. These can also be used as consistencies. Note that the number of spaces is not consistent, since there are four spaces in rows 2 and 3, but only three spaces in rows 4 and 5; though counting from right to left, there is a consistency in the number of spaces to the beginning of the ID number, and that same consistency identifies the end of the name field.
3. There is a single space that precedes the email address for everyone. If one could locate the position of that space, one would be able to extract the email.
4. There is also a single @ sign in each row. The @ sign could be used to separate the first part of the email from the domain part.

The Excel function (equation 1) to extract only the student name is

=LEFT(A2,23)	(1)
=LEFT(A2,FIND(",",A2)-1)	(2)

Recall that this function requires two elements in the argument, the location of the text from which the subset of characters will be taken (cell A2) and the number of characters to take (23 characters). In this example, the formula is to extract the substring from the content in cell A2, then the remainder of the cells in column A, once the formula is copied down. The number of characters to extract is, however, not consistent from one cell to the next. The name in row 2 contains 23 characters but the one in row 3 contains 17 characters. Therefore, one must use a formula that will evaluate to the needed number of characters. To calculate the number of characters to extract from each cell in the column of data, it is necessary to utilize some consistency located in the area of the end of the name portion of the string. In this example, that consistency is the location of the first comma located in position 24; that is, it is the 24^{th} character counting from left to right in A2. To find the location of any character or set of characters in a string, the FIND() function is used:

=FIND(the character or set of characters, the cell to be searched)

The character to be located is the comma, ",", and the cell to search is A2. Therefore, the completed formula is =FIND(",",A2). The completed function is shown in (2). This returns the value 24. However, 24 is not 23, so it is necessary to perform one other operation. A question to ask of students seems trivially obvious, but it needs to be addressed, is: how to convert the 24 into 23. Of course, the answer is to subtract one, but the arithmetic operations around string functions are not yet internalized. Thus it is somewhat of a realization to discover that one can subtract a number from a calculated value.

The formula in (2) evaluates to =LEFT(A2,24-1) or =LEFT(A2,23). If the length of all the names were consistently 23 characters, one could simply place the number 23 into the second position in the argument; but the number of characters varies from the name in row 2 to row 3, and so forth. To better explain the nesting of

functions, the decomposition/reconstruction method is shown diagrammatically:

Level 1	=LEFT(----------)
Level 2	A2,FIND(-------)-1 (3)
Level 3	",",A2

The uppermost level the driving function, LEFT in this case, is shown with a dashed line that represents the contents of its arguments. At the second level are the contents, including another function, FIND. It, too, has an argument which is shown on yet another level. At this third level there are no other functions. Therefore, at the conclusion of the reading of the function at the bottom level, one jumps up to Level 2, where the FIND argument concludes with the remaining value that is not itself a function. After the "-1", the remaining function is concluded at Level 1.

To reconstruct the function, such that the student can easily apprehend how the functions work, substitute the evaluated value at each level.

Level 1	=LEFT(A2,23)
Level 2	A2,24-1 (4)
Level 3	The comma is the 24th character in the string.

Level 3 has a few words that describes the evaluation of those characters, in this case that the "," is located in position 24 in the string. In level 2 the FIND(----------) is replaced by 24 and the remainder of the contents in the argument are placed there. Finally the values of level 2 are placed in their location in level 1 and the student can now understand that 23 characters (Bigham ToughGuy Hotshot) are being copied beginning at the left end of the string. It should be easy to see that as the formula is copied down to the next cell the reference to A2 changes to A3, the evaluation of the location of the first comma is different, but the entire student name is captured.

The next item to capture is the ID number. This is merged with the cell phone number (note that the example is limited to North American phone numbers for consistency in the length of the number) but in North America it is known that phone numbers contain 10 digits and that there are 20 digits in each row. Therefore, this is a consistency that can be used in identifying the number of characters to extract. For all substrings contained within another string that do not begin with the first character nor end with the last character, the function of choice is the MID command. The parts of this function are:

MID(the cell containing the string from which you want to extract a substring, the starting point for the extraction, the number of characters to extract).

Consequently, the formula that will extract the ID number is MID(A2,26,10). The last part of the argument, 10, is the number of characters to extract for our substring. For this example, the position of the "6" is the 36th character from the left. The only part of this formula that changes as it is copied down from A2 to A5 is the starting point, 26 in this example. To place the correct number into each evaluation of the succeeding formulas requires a mechanism to identify that beginning point. Carrying out this task requires the identification of some consistency in the formula, and it is identified with the location of the first comma. Now all the consistencies have been identified and it is known that the first comma is the closest consistency to the beginning of the start position. Therefore, one must find the comma and add two to that result, since the starting position is consistently located two characters past the first comma. The formula that correctly captures the substring is

=MID(A2,FIND(",",A2)+2,10)

and it is decomposed in (6).

Level 1	=MID(----------)
Level 2	A2,FIND(----)+2,10 (6)
Level 3	",",A2

This structure appears very familiar since something very close to what was just used for the student name. The function is shown in level 1 and the contents of its argument are at level 2. Since there is another function there, the FIND, its argument is displayed in level 3.

The formula is now reconstructed:

Level 1	MID(A2,26 ,10)
Level 2	A2,24+2,10 (7)
Level 3	The comma is the 24th character in the string.

The final form of the formula in level 1 is the correct equation that Excel evaluates.

The third part of the extraction is that of the cell phone number. The consistencies identified in the data permit us to observe that the cell phone number ALWAYS begins at the 11th position **following the beginning** of the ID number and that it is ALWAYS 10 characters long (see (8)). For the string in cell A2, the correct answer is =MID(A2,36,10) but the starting position is not consistent with 36, as can be seen, if one counts over in cell A3, so a formulation must be made to capture that value. Again, it is noted in the search for consistencies that only the first comma can be exploited, augmented with the addition of 12, to arrive at the start position. The number 12 is made up of the 2 that gets us to the start of the ID number and the remaining 10, whichis the length of thecell number: 10 + 2 = 12. The start position for the cell phone number is consistently 12 characters after the first comma. To construct the solution it is necessary to first decompose the problem (9):

The evaluated function is =MID(A2,36,10). The elements in the argument that are constant when the formula is copied down to cell A5 are the cell, A2—that automatically increments, because it is a relative cell address—and the number of characters to extract—10 in this case, because all cell phone numbers in column A are 10 characters in length. What does change, however, is the starting position for the data extraction. In cell A2 it is 36, in A3 it is 30, and in A4 it is 31. Since these numbers are inconsistent, some other method must be used to identify the start position.

It can be observed that the only consistent element that precedes the phone number is the first comma, so if the location of the comma can be found and the number 12 added to it, the correct starting position for all the cell phone numbers will have been identified. The formula is =MID(A2,FIND(",",A2)+12,10). To visualize this formula to make it easier for students to understand, it is decomposed into its corresponding levels (figure 8):

Level 1	=MID(----------)
Level 2	A2,FIND(----)+12,10 (9)
Level 3	",",A2

The similarity between (6) and (9) is remarkable; the student should be quick to observe that the 2 of (7) is now the 12 of (9). It is the starting position alone that has changed!

The decomposition in (9) is now reconstructed in (10):

Level 1	=MID(A2,36 ,10)
Level 2	A2,24+12,10 (10)
Level 3	The comma is the 24th character in the string.

Finally, the last part of the extraction is the email address, which is the last item in A2. This fact permits us to use the RIGHT function, =RIGHT(cell from which to extract characters from the right side, the number of characters to extract).

Looking at A2, the completed solution is =RIGHT(A2,18), for A3 it is =RIGHT(A2,17) and A5 is =RIGHT(A2,22). What changes is the number of characters to extract. Again, the solution begins with an inspection of the consistencies, and it is noted that the presence of a comma (again) is just before the email address. It is the *second* comma, so a simple FIND(",",A2) will never return the position of the second comma, since it is the first comma that will intercept the evaluation. Here is an obvious fact that seems sometimes to evade the perception of UGs: The FIND() function ALWAYS evaluates a string from LEFT to RIGHT. Merely nesting the function within the RIGHT() function does nothing to change the control of the FIND() function: it always evaluates from left to right. Thus, students seem to think that a logical solution to the problem is a simple =RIGHT(A2,FIND(",",A2)). In fact, this evaluates to =FIND(A2,24) and not the desired =FIND(A2,18). The following figure should help clarify this specific situation:

To even consider a formula, one must be able to visualize these positions at least in one's mind. It is not unusual for students to bypass thought and go straight to typing a solution that is usually incorrect. To solve this problem, it is noted that one of the needed values appears missing from (11), the ending position. Fortunately, Excel has a function that will always calculate the number of characters in a string, LEN().

[A short tangent is in order here. If spurious spaces are contained in the string, that is, spaces before the first character, spaces after the last character, and/or multiple spaces with the string, the TRIM() function will remove all leading and trailing spaces and reduce the aggregate spaces in a group to only one space

per group. Hence, students should be told to trim all the cells with =TRIM(cell) then copy that entire block of cells and use "paste special": "values", in order to paste only the values over the top of the TRIM() formulas. The original data can then be deleted and students are able to work with strings in which they know there are no leading, trailing, or extra spaces in the string. Now the student can continue with the extraction of the email address.]

The =LEN(A2) formula gives a value of 65 and LEN(A3) is 58. The function RIGHT takes only the cell of interest and the number of characters to extract from the ***right***. Recall that FIND provides a value ONLY when counted from the left end of the string, so RIGHT(A2,FIND(",",A2)) produces a very wrong answer, since it evaluates to RIGHT(A2,24), and this yields the email address "1234, bhotshot@gmail.com"! Extracting the proper number of characters from the right side of a string requires the use of a special technique.

The general solution to substring extraction from the right is to determine the number of characters in the entire string, then subtract from that the number of characters one wishes to omit from the extraction. For this example, the number of characters is 65 and it is desirable to omit everything before the email address. The consistency at that location—the area in the beginning of the email address—is a bit harder to determine. One must either find a consistency or create one. Indeed, there is a consistency, in that the comma-space before the address is the second such combination in the string. There are two methods that can use to specifically identify the second comma. The first method is the FIND() function.

In its extended form, FIND(character(s) sought, cell that contains the string), an optional argument item is available that has the FIND function beginning its search at some predetermined location in the string. In the following Table 1 one can see the basic FIND seeking to locate the first comma, with a result of 24. On row 2 is the same FIND, but with the optional component of identifying a position in the string from which the function is to **begin** its search for the comma; in this case the position is 25. Any other number less than or equal to 24 would result in the function returning 24, because that would be the first comma the function meets. Looking down the column of cells, it is apparent that one cannot simply append a number as the third argument. One way to solve the problem is to employ another FIND() to calculate the position in the string with which to begin the search for the comma. Rows 3 and 4 show a FIND function nested within a FIND() function that yields the correct location of the second comma. Using a nested FIND for the third instance of a character, say a third comma, becomes confusing, but there is a better solution.

Row	Formula	Result
1	=FIND(",",A2)	24
2	=FIND(",",A2,25)	46
3	=FIND(",",A2,FIND(",",A2)+1)	46
4	=FIND(",",A3,FIND(",",A3)+1)	24

Table 1 Nesting a FIND within a FIND.

The SUBSTITUTE() function has many very useful applications. Its format is

=SUBSTITUTE(cell containing the string, the old character(s) you wish to replace, the new character(s), and an optional part that identifies which the old character(s) is to be replaced.)

If the optional component is omitted, then all old characters are replaced with the new characters. In other words, one character is simply replaced with another. On the surface this appears rather unimportant, but in practice it allows us to solve certain problems that otherwise would be far more difficult, if not impossible, using a spreadsheet.

One can now see that engaging in string manipulation requires one to take advantage of data consistencies, but sometimes there are simply no consistencies usable for the task at hand. In those situations it is necessary to create a consistency, and it is the SUBSTITUTE function that accomplishes this task. This is explained using the current example. If the string in A2 were to be represented as (12) with the second comma replaced with the pipe symbol "|",then the position of the "|" can be easily found with a simple FIND(). The solution to calculating the number of characters to extract the email address is LEN(A2)-FIND("|",A2)-1 and the decomposition of the entire equation is:

```
=RIGHT(A2,LEN(A2)-FIND("|",A2)-1)

          A2,LEN(---)-FIND(---------)-1)        (13)
                A2          "|",A2
```

ISSN: 2167-1435
v31 n3005

The reconstruction is shown as:

So there is a solution to the problem of extracting the right-most element in the string, the email address. But the question now remains as to HOW does a pipe symbol become part of the string? Recall that while the second comma is a consistency, it is not so easy to find its location in the string, hence the need to be creative in the solution. This is the time to consider placing some unique character in the string so that the FIND can locate it.

The SUBSTITUTE function is ideal for this task. There are two locations where a string can be placed while Excel works upon it: one place is a cell on the worksheet, the other is RAM, where the modified string exists only while Excel is evaluating it, and afterwards it disappears. If the former location is used, the worksheet will quickly fill up with unnecessary and duplicated data that can easily cause confusion and lead to errors. The latter location creates a string in memory, manipulates it as necessary, and evaluates it to produce the result that is brought into another formula. The latter is the much preferred way.

The instructor can show how both of these methods work and why one is preferred over the other by using the following technique to place a special character into a string, creating a consistency where none previously existed. Figure 14 shows the formula to place the pipe symbol, "|", where the second comma is located.

The cell on which the substitution is to occur is A2, the character to replace is one of the commas, the replacement character is the pipe, and it is the second comma ONLY on which the replacement is performed. The string then becomes as shown in (12) above. It should be obvious that the pipe replaced the second comma. If the student were to place each of these cells into a companion column, the meaning of the data would be unchanged. However, the point of creating a consistency is to discover the location of the pipe in the string and to use that answer elsewhere. Recall that the FIND function is used to return the location of a character, so one need only to nest the SUBSTITUTE function within the FIND function and the number is returned without the need for the intermediate string step. The result is

=FIND("|",SUBSTITUTE(A2,",","|",2))

which evaluates to

=RIGHT(A2,18) A2,65 - 46 -1) The number of characters in the string and the location of the pipe symbol.	(14)

=FIND("|","Bigman ToughGuy Hotshot,
10000234566072741234|
bhotshot@gmail.com").

It is easy to see that this evaluates to 46. The "B" in Bigman is character #1, the "H" in Hotshot is character #17, and the value of the "|" is 46. One needs only the number, not the string with the pipe, so it is preferable not to display the modified string in a cell.

Now that there is a way of identifying a consistency that occurs before the email address, and since it is already known the number of characters to the end of the address, a simple subtraction gives the number of characters to extract from the right side. The resulting formula is

=RIGHT(A2,LEN(A2)-
FIND("|",SUBSTITUTE(A2,",","|",2)-1)

This is more easily understood, when rewritten as a decomposed formula, as in Figure 15.

… followed by the reconstruction from the bottom up.
It is now easy to see how functions are nested, evaluated, and how they yield the desired output.

4. CONCLUSIONS

The instructor can more clearly illustrate and explain how to manipulate string functions to extract data that are not separable using the text-to-columns method. As expertise develops in this area of spreadsheet mastery, the user will find that the cleansing of data files for importation into database tables can be far more easily facilitated, and the instructor will have a tool by which students can develop skill in mastering how to think in a sequential manner, for these spreadsheet functions are evaluated from the inside out and the student must decide what data need to be acquired, before they can be manipulated. It can be argued that the only other alternative way of developing this skill is to learn how to program. Programming develops thinking skills, but using these string

ISSN: 2167-1435
v31 n3005

manipulation functions is as close as one can get to achieve these skills, when programming is not an already available skill.

Instructors can create complex functions—that is, nested functions that could draw data from differing parts of a worksheet—and explain them by decomposing a function to its basic parts, providing answers to the simple parts, and reconstructing the formula with values that lead to the desired result. In this way, students become far more comfortable with sequential thinking, the nesting of spreadsheet functions, and the flow of control in their formulas.

4. REFERENCES

Jonassen, D., Carr, C., & Yueh, H. (1998). Computers and mindtools for engaging learners in critical thinking, *TechTrends*, Springer.

Caulkins, J., Morrison, E., & Weidemann, T. (2007). Spreadsheet Errors and decision making: Evidence from field interviews, *Journal of Organizational and End User Computing*.

Carrithers, D., Ling, T. & Bean J. (2012). Messy problems and lay audiences: Teaching critical thinking within the finance curriculum, *Business and Professional Communication Quarterly*.

ISSN: 2167-1435
v31 n3005

Figures

	A
1	**Student_Name, IDNumCellNum, Email**
2	Bigman ToughGuy Hotshot, 10000234566072741234, bhotshot@gmail.com
3	Phrank Lee Spider, 77777445553181234567, pspider@yahoo.com
4	Sindiee Catapeller, 66666711221213456789, scata@hotmail.com
5	Mamoud Tinkelpheater, 44444556993735648765, mtinkel@macmedpetra.ca

Figure 5

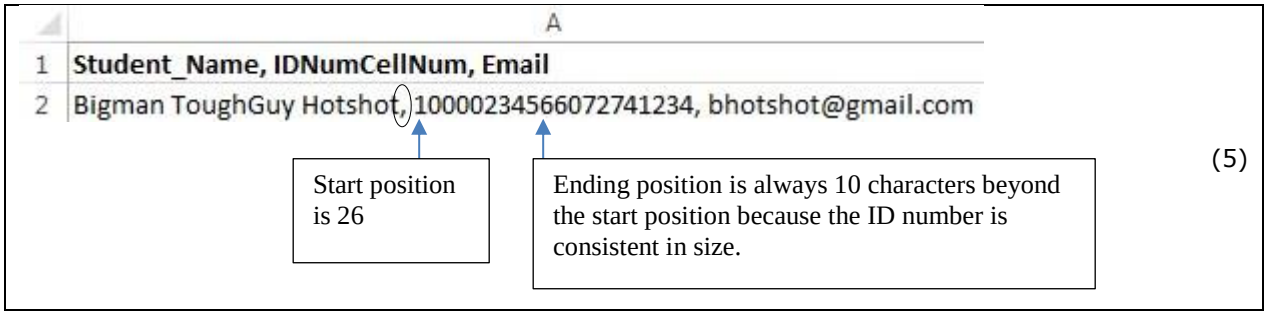


Figure 8

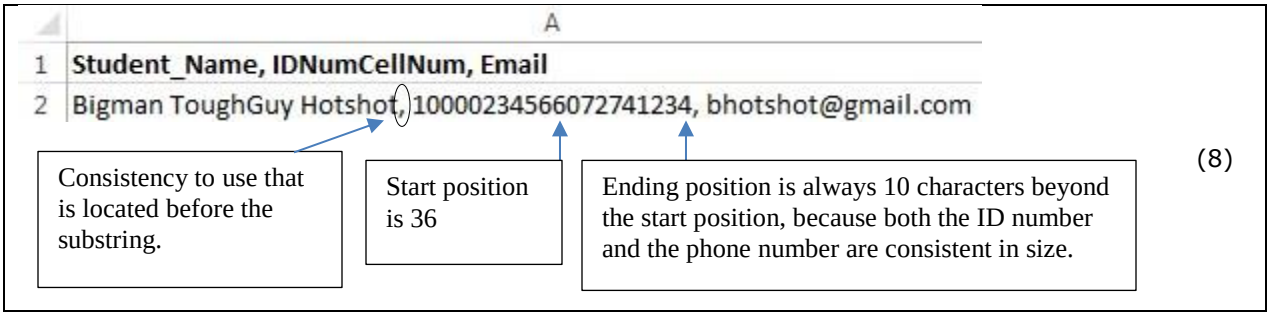
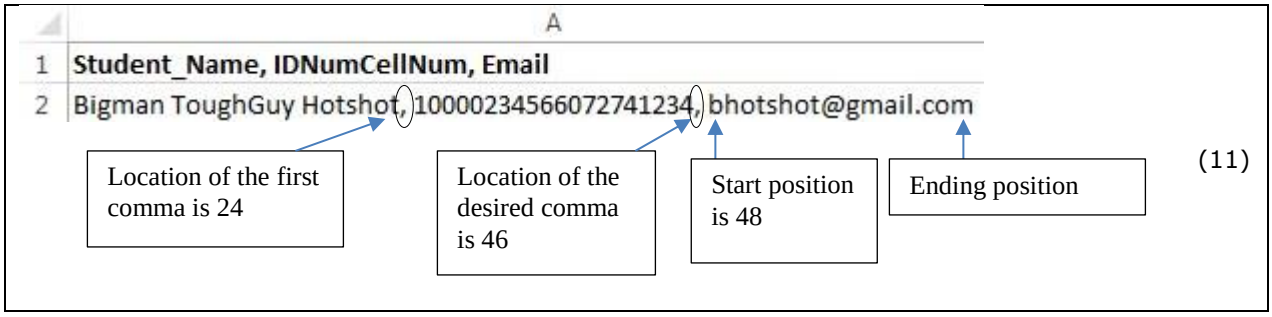


Figure 11



ISSN: 2167-1435
v31 n3005

Figure 12

Bigman ToughGuy Hotshot, 10000234566072741234| bhotshot@gmail.com	(12)

Figure 15

Bigman ToughGuy Hotshot, 10000234566072741234, bhotshot@gmail.com
Bigman ToughGuy Hotshot, 10000234566072741234\| bhotshot@gmail.com

=SUBSTITUTE(A2,",","\|",2) (15)

Old character to replace

New character to take the place of the old character.

Which instance of the old character to replace.

Figure 16

```
=RIGHT(-------------------------------------------------- -1)
          A2,LEN(--) - FIND(--------------------------------
                A2          "|",SUBSTITUTE(-------------)
                                          A2,",","|",2
```
(16)

Figure 17

```
=RIGHT(A2,65-46-1) OR RIGHT(A2,18)
          A2,65 - 46
"|","Bigman ToughGuy Hotshot, 10000234566072741234| bhotshot@gmail.com")
                                  Replace the second comma with the pipe.
```
(17)