Bug#1

1. The line(s) containing each bug:



1. The corrected version of the line(s):



1. What was wrong with the line(s) and how you fixed it:

This place should increment *numLines* after assigning value to *lineStarts* array. I added a *++* increment operator to *numLines* to ensure that line number is assigned at the correct position from the next iteration.

1. The debugger tool or technique you used to recognize and find the bug:

I placed a breakpoint on the lines containing the bug and find out the value was assigned at the same position every iteration.

Bug#2

1. The line(s) containing each bug:



1. The corrected version of the line(s):



1. What was wrong with the line(s) and how you fixed it:

This place didn’t handle the situation the input string str containing null terminator at the position i. I added a condition to check this situation. If it happens, it will terminate the for-loop and leading to return result immediately.

1. The debugger tool or technique you used to recognize and find the bug:

I find out this bug with my observation.

Bug#3

1. The line(s) containing each bug:



1. The corrected version of the line(s):



1. What was wrong with the line(s) and how you fixed it:

The code after this line would consume the result from the *nextWhite* function, and that could return -1 value. If that happens, it would cause a unexpected behavior, so I put a condition to this place to catch -1 value and replace it with a value of *strlen(str + i)*.

1. The debugger tool or technique you used to recognize and find the bug:

I find out this bug with my observation.

Bug#4

1. The line(s) containing each bug:



1. The corrected version of the line(s):



1. What was wrong with the line(s) and how you fixed it:

This original calculation is wrong, and it wasn’t the length of the word or number. I fixed it with the value of *sp* to be the correct length of the word or number.

1. The debugger tool or technique you used to recognize and find the bug:

I placed a breakpoint on the line and found out the original *(sp -i + 1)* wasn’t indicate the length of the word

Bug#5

1. The line(s) containing each bug:



1. The corrected version of the line(s):

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

1. What was wrong with the line(s) and how you fixed it:

The code after this line needs to use *sp* to iterate, but it could be -1 as it is the result from the *nextWhite* function. Thus, I use an if-condition to catch -1 value and reassign *sp* with a correct one.

1. The debugger tool or technique you used to recognize and find the bug:

I find out this bug with my observation.

Bug#6

1. The line(s) containing each bug:



1. The corrected version of the line(s):



1. What was wrong with the line(s) and how you fixed it:

The place used an incorrect indexing number to access the original string, so I instead use *(start + 1)* to access the correct position of that character.

1. The debugger tool or technique you used to recognize and find the bug:

I placed a breakpoint on the line and found out the original accessing position should not be *i*.

Bug#7

1. The line(s) containing each bug:



1. The corrected version of the line(s):

None

1. What was wrong with the line(s) and how you fixed it:

This place decremented the *numWord* by one unnecessarily. I just simply removed it.

1. The debugger tool or technique you used to recognize and find the bug:

I placed a breakpoint to see the value of *numWord* every time using *getWord* function and found out *numWord* will be decremented by 1.

Bug#8

1. The line(s) containing each bug:



1. The corrected version of the line(s):

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

1. What was wrong with the line(s) and how you fixed it:

The place missed a check to handle the boundary -1 value. Thus, I added one and reassign *sp* with a correct one.

1. The debugger tool or technique you used to recognize and find the bug:

I found out this bug with my observation.