**Workshop 6**

Name : Ahram Lee

Email : alee239@myseneca.ca

Student ID : 133849232

Date: 2024-06-15

**Bug 1**

**Lines Containing the Bug** :

if (str[i] == ' ' || str[i] == '\t') {

result = i;

}

**Corrected Version** :

if (isspace(str[i])) {

result = i;

}

**What Was Wrong and How you fixed it**:

The original code only checked for spaces and tab characters, but there are other whitespace characters like newlines that need to be considered. The isspace function was used to check for any whitespace character, making the function more robust.

**Debugger Tool/Technique Used** :

Used assertions and log statements to identify that the function did not handle all whitespace characters correctly.

**Bug 2**

**Lines Containing the Bug** :

for (result = 1; result <= len && result; result++) {

if (!isdigit(str[result - 1])) result = -1;

}

**Corrected Version** :

for (i = 0; i < len && result; i++) {

result = result && isdigit(str[i]);

}

**What Was Wrong and How you fixed it**:

The original loop incorrectly iterated and checked the condition for being a number. The corrected version uses a proper loop structure to check each character up to the specified length, ensuring all characters are digits.

**Debugger Tool/Technique Used** :

Verified the output using log statements and added assertions to ensure the function correctly identifies numeric strings.

**Bug 3**

**Lines Containing the Bug** :

while (str[i] != '\0' && isspace(str[i])) {

if (str[i] == '\0') {

result.lineStarts[result.numLines++] = i + 1;

}

i++;

}

**Corrected Version** :

while (str[i] != '\0' && isspace(str[i])) {

i++; // Skip over any whitespace characters

}

if (str[i] == '\n') {

result.lineStarts[result.numLines++] = i + 1;

}

**What Was Wrong and How you fixed it**:

The original code incorrectly handled line breaks within the loop for skipping whitespace characters. This resulted in incorrect line counting..

**Debugger Tool/Technique Used** :

Used assertions to check the number of lines and log statements to trace the index creation process.

**Bug 4**

**Lines Containing the Bug** :

word[sp] = '\0';

**Corrected Version** :

f word[sp - start] = '\0';

**What Was Wrong and How you fixed it**:

The position of the null terminator was incorrectly set, which could cause incorrect string lengths or buffer overflows. The correction ensures the null terminator is placed at the correct position relative to the start.

**Debugger Tool/Technique Used** :

Assertions to check string lengths and manual inspection of output strings to ensure they were correctly terminated.