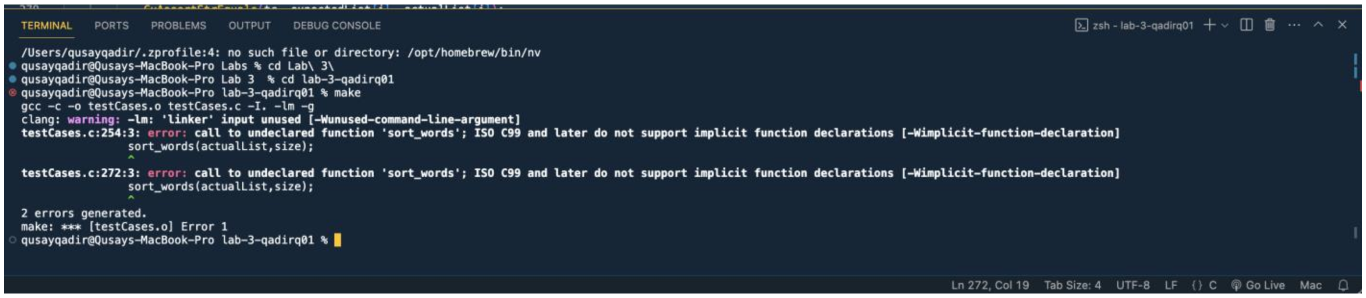
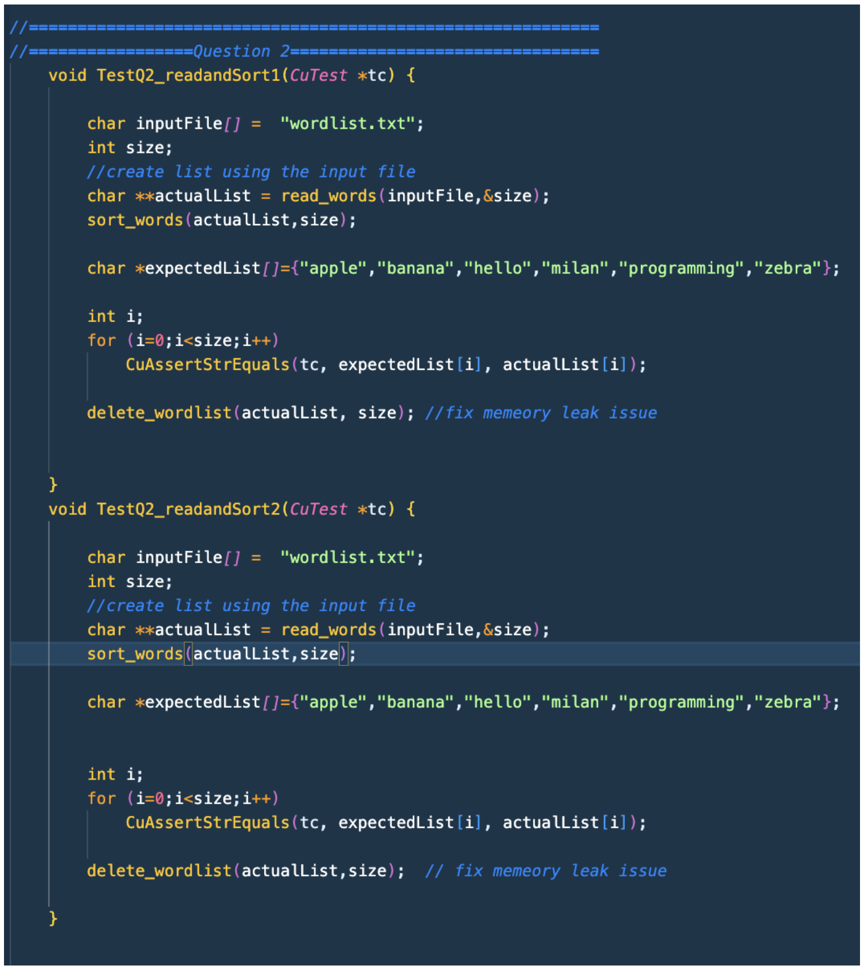
Bug Observation #1:

- Code runs, but there is an issue with the test case file calling the wrong functions, that causes the complier to throw an error.



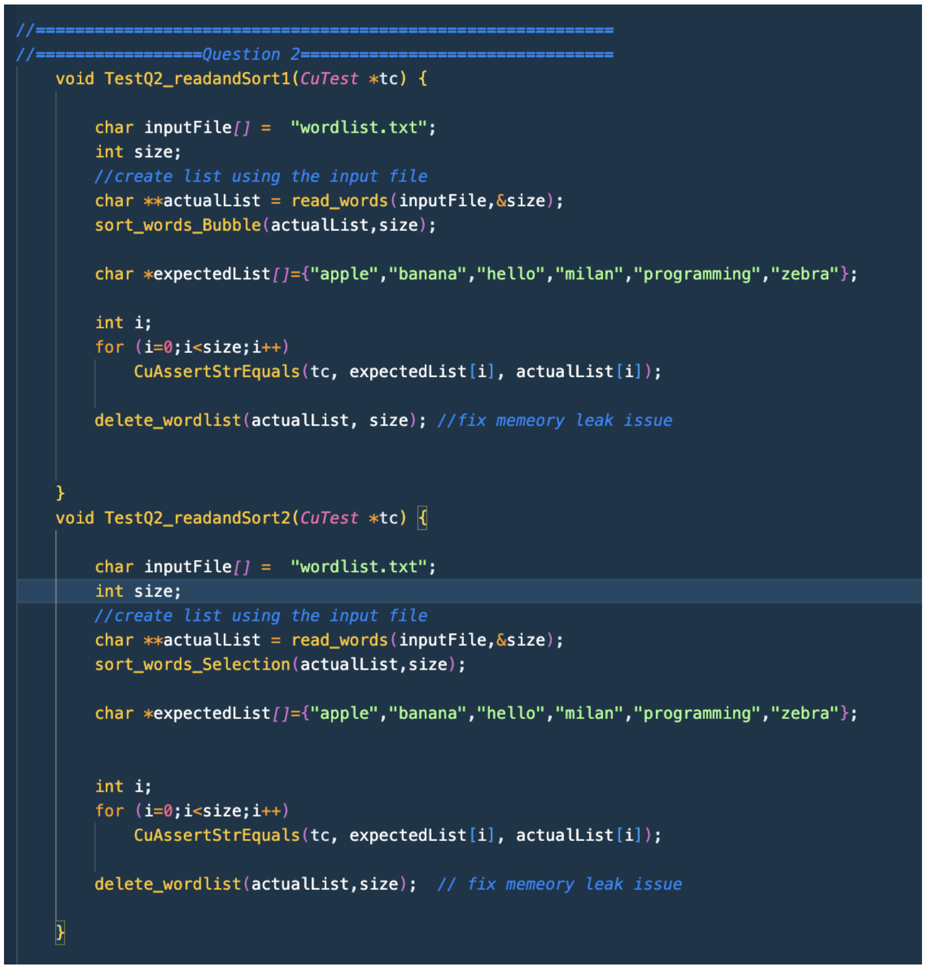
```
/Users/qusayqadir/.zprofile:4: no such file or directory: /opt/homebrew/bin/nv
qusayqadir@Qusays-MacBook-Pro Labs % cd Lab\ 3\
qusayqadir@Qusays-MacBook-Pro Lab 3  % cd lab-3-qadirq01
qusayqadir@Qusays-MacBook-Pro lab-3-qadirq01 % make
gcc -c -o testCases.o testCases.c -I. -lm -g
clang: warning: -lm: 'linker' input unused [-Wunused-command-line-argument]
testCases.c:254:3: error: call to undeclared function 'sort_words'; ISO C99 and later do not support implicit function declarations [-Wimplicit-function-declaration]
                sort_words(actualList,size);
                ^
testCases.c:272:3: error: call to undeclared function 'sort_words'; ISO C99 and later do not support implicit function declarations [-Wimplicit-function-declaration]
                sort_words(actualList,size);
                ^
2 errors generated.
make: *** [testCases.o] Error 1
qusayqadir@Qusays-MacBook-Pro lab-3-qadirq01 %
```

Bug Fix Validation #1:



```c
//==============================================================
//================Question 2===============================
    void TestQ2_readandSort1(CuTest *tc) {

        char inputFile[] =  "wordlist.txt";
        int size;
        //create list using the input file
        char **actualList = read_words(inputFile,&size);
        sort_words(actualList,size);

        char *expectedList[]={"apple","banana","hello","milan","programming","zebra"};

        int i;
        for (i=0;i<size;i++)
            CuAssertStrEquals(tc, expectedList[i], actualList[i]);

        delete_wordlist(actualList, size); //fix memeory leak issue


    }
    void TestQ2_readandSort2(CuTest *tc) {

        char inputFile[] =  "wordlist.txt";
        int size;
        //create list using the input file
        char **actualList = read_words(inputFile,&size);
        sort_words(actualList,size);

        char *expectedList[]={"apple","banana","hello","milan","programming","zebra"};


        int i;
        for (i=0;i<size;i++)
            CuAssertStrEquals(tc, expectedList[i], actualList[i]);

        delete_wordlist(actualList,size);  // fix memeory leak issue

    }
```

- The code that was causing the complier error ^

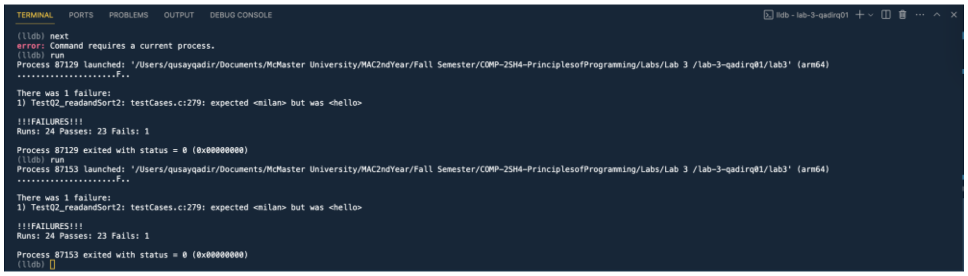


```c
//============================================================
//=================Question 2================================
    void TestQ2_readandSort1(CuTest *tc) {

        char inputFile[] =  "wordlist.txt";
        int size;
        //create list using the input file
        char **actualList = read_words(inputFile,&size);
        sort_words_Bubble(actualList,size);

        char *expectedList[]={"apple","banana","hello","milan","programming","zebra"};

        int i;
        for (i=0;i<size;i++)
            CuAssertStrEquals(tc, expectedList[i], actualList[i]);

        delete_wordlist(actualList, size); //fix memeory leak issue


    }
    void TestQ2_readandSort2(CuTest *tc) {

        char inputFile[] =  "wordlist.txt";
        int size;
        //create list using the input file
        char **actualList = read_words(inputFile,&size);
        sort_words_Selection(actualList,size);

        char *expectedList[]={"apple","banana","hello","milan","programming","zebra"};


        int i;
        for (i=0;i<size;i++)
            CuAssertStrEquals(tc, expectedList[i], actualList[i]);

        delete_wordlist(actualList,size);  // fix memeory leak issue

    }
```

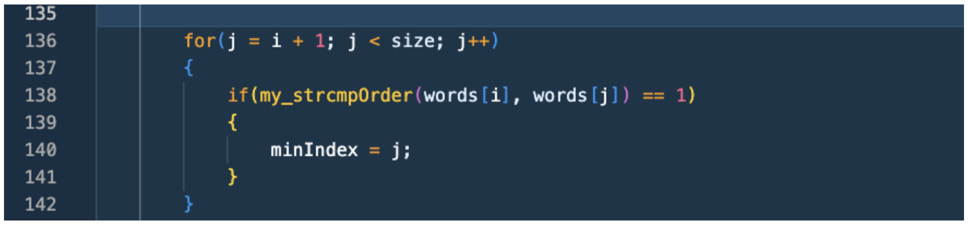
- Code that fixes the complier error ^

Bug Observation #2:

- Seems to be a problem with the word swapping indexes



```
(lldb) next
error: Command requires a current process.
(lldb) run
Process 87129 launched: '/Users/qusayqadir/Documents/McMaster University/MAC2ndYear/Fall Semester/COMP-2SH4-PrinciplesofProgramming/Labs/Lab 3 /lab-3-qadirq01/lab3' (arm64)
.....................F..

There was 1 failure:
1) TestQ2_readandSort2: testCases.c:279: expected <milan> but was <hello>

!!!FAILURES!!!
Runs: 24 Passes: 23 Fails: 1

Process 87129 exited with status = 0 (0x00000000)
(lldb) run
Process 87153 launched: '/Users/qusayqadir/Documents/McMaster University/MAC2ndYear/Fall Semester/COMP-2SH4-PrinciplesofProgramming/Labs/Lab 3 /lab-3-qadirq01/lab3' (arm64)
.....................F..

There was 1 failure:
1) TestQ2_readandSort2: testCases.c:279: expected <milan> but was <hello>

!!!FAILURES!!!
Runs: 24 Passes: 23 Fails: 1

Process 87153 exited with status = 0 (0x00000000)
(lldb)
```



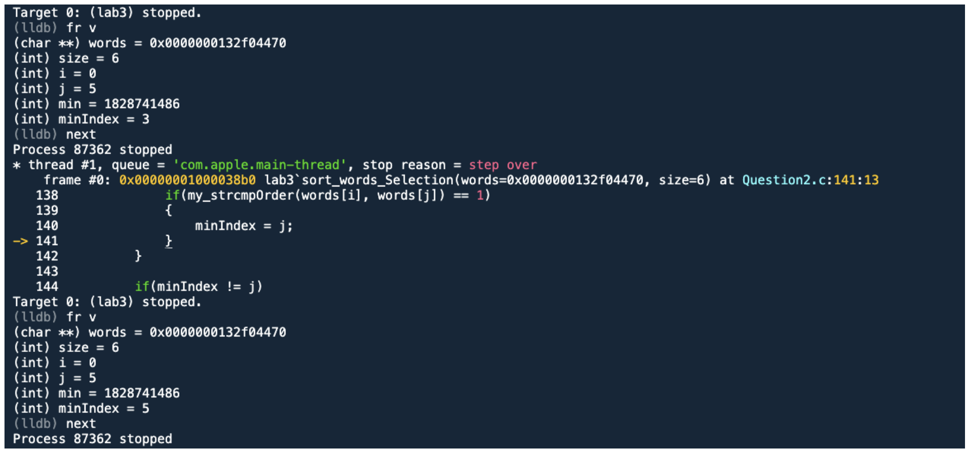
```
135
136         for(j = i + 1; j < size; j++)
137         {
138             if(my_strcmpOrder(words[i], words[j]) == 1)
139             {
140                 minIndex = j;
141             }
142         }
```

GBD Anaylsis #2:

- Create a breakpoint at line 136, because the for loop is where the strings are being indexed and compared to each other.

Possible Root Cause:

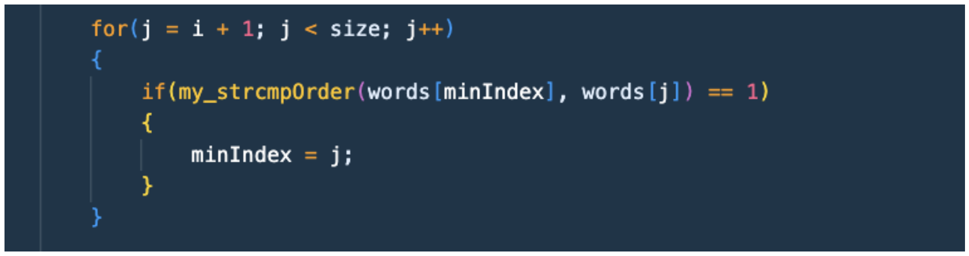
- The comparison should be with the current element and the new minindex that is being set,



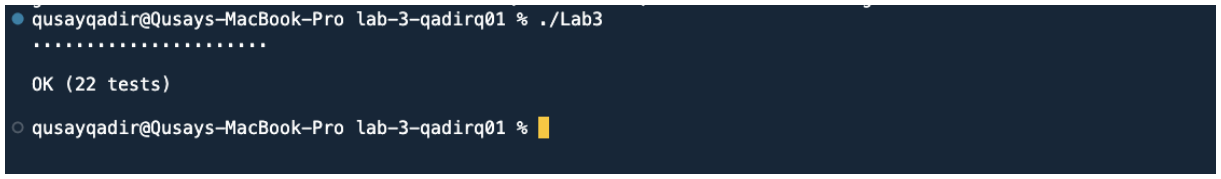
```
Target 0: (lab3) stopped.
(lldb) fr v
(char **) words = 0x0000000132f04470
(int) size = 6
(int) i = 0
(int) j = 5
(int) min = 1828741486
(int) minIndex = 3
(lldb) next
Process 87362 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = step over
    frame #0: 0x00000001000038b0 lab3`sort_words_Selection(words=0x0000000132f04470, size=6) at Question2.c:141:13
   138                  if(my_strcmpOrder(words[i], words[j]) == 1)
   139                  {
   140                      minIndex = j;
-> 141                  }
   142              }
   143
   144              if(minIndex != j)
Target 0: (lab3) stopped.
(lldb) fr v
(char **) words = 0x0000000132f04470
(int) size = 6
(int) i = 0
(int) j = 5
(int) min = 1828741486
(int) minIndex = 5
(lldb) next
Process 87362 stopped
```

The above shows that placing the breakpoint at line 136 and stepping through them while printing the local variables show there is an error with the change of min index.

Bug Fix Validation #2:



```
    for(j = i + 1; j < size; j++)
    {
        if(my_strcmpOrder(words[minIndex], words[j]) == 1)
        {
            minIndex = j;
        }
    }
```

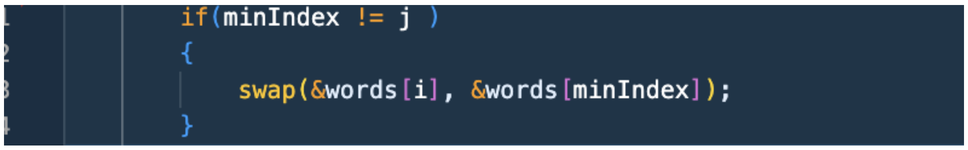


```
qusayqadir@Qusays-MacBook-Pro lab-3-qadirq01 % ./Lab3
......................

OK (22 tests)

qusayqadir@Qusays-MacBook-Pro lab-3-qadirq01 %
```

- Code runs without error and passes all test cases

Bug Observation #3:

- There seems to be a logical flaw with this block of code,

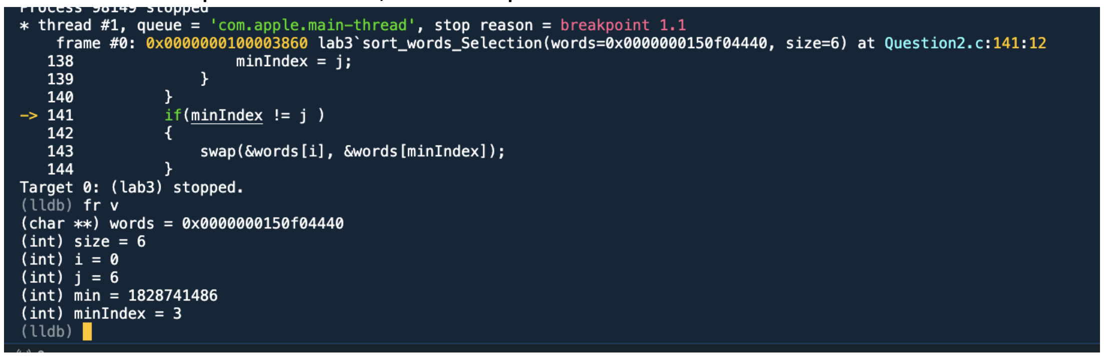


```
        if(minIndex != j )
        {
            swap(&words[i], &words[minIndex]);
        }
```

The error here shows that minIndex will always be compared to the value of j and since this statement is outside the inner for loop the value of j will always be 6, so the minindex will always be compared to the last element instead of comparing it to the current assumed min index value which is i.

GBD Analysis #3:

- Create a break point at value, and compare the variable values.

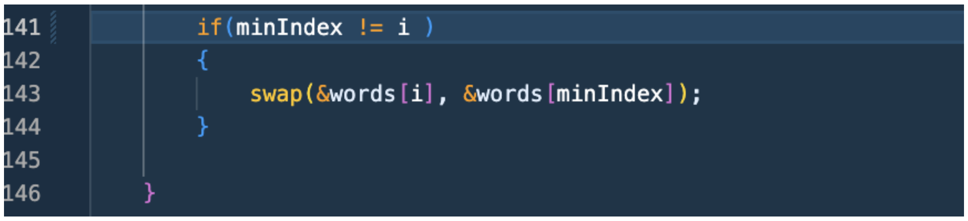


```
Process 98149 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
    frame #0: 0x0000000100003860 lab3`sort_words_Selection(words=0x0000000150f04440, size=6) at Question2.c:141:12
   138                      minIndex = j;
   139                  }
   140              }
-> 141              if(minIndex != j )
   142              {
   143                  swap(&words[i], &words[minIndex]);
   144              }
Target 0: (lab3) stopped.
(lldb) fr v
(char **) words = 0x0000000150f04440
(int) size = 6
(int) i = 0
(int) j = 6
(int) min = 1828741486
(int) minIndex = 3
(lldb) 
```

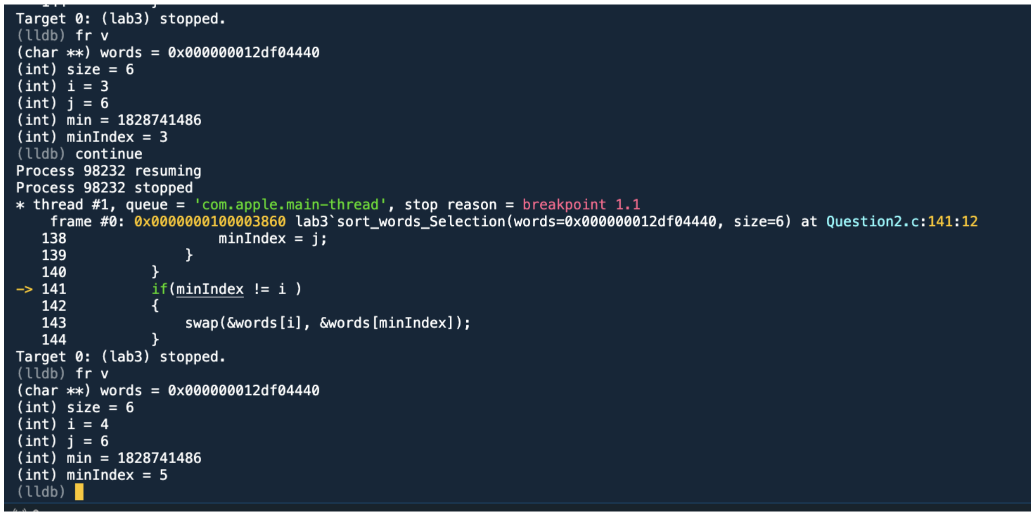
Possible Root Cause:

- The comparison of the minindex, should be with I because after the inner for loop is ran it should check weather there is a new minindex (the value of j) that is should be swapped with.

Bug Fix Validation #3:



```
        if(minIndex != i )
        {
            swap(&words[i], &words[minIndex]);
        }

    }
```



```
Target 0: (lab3) stopped.
(lldb) fr v
(char **) words = 0x000000012df04440
(int) size = 6
(int) i = 3
(int) j = 6
(int) min = 1828741486
(int) minIndex = 3
(lldb) continue
Process 98232 resuming
Process 98232 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
    frame #0: 0x0000000100003860 lab3`sort_words_Selection(words=0x000000012df04440, size=6) at Question2.c:141:12
   138                     minIndex = j;
   139                 }
   140             }
-> 141             if(minIndex != i )
   142             {
   143                 swap(&words[i], &words[minIndex]);
   144             }
Target 0: (lab3) stopped.
(lldb) fr v
(char **) words = 0x000000012df04440
(int) size = 6
(int) i = 4
(int) j = 6
(int) min = 1828741486
(int) minIndex = 5
(lldb)
```

- Code runs without errors passes all test cases, and satisfies the logic of the selection sort and how the values of j and I correctly are being associated with the change and comparison of minIndex.