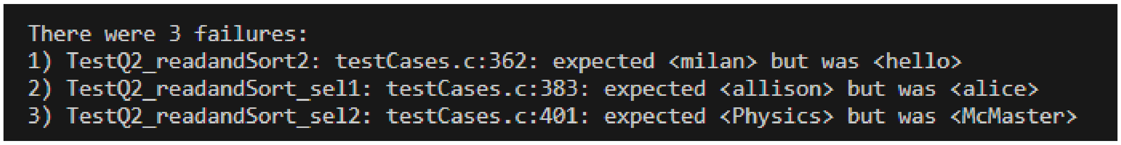
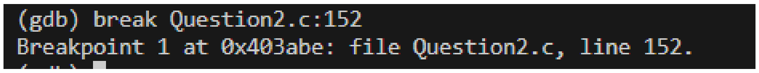
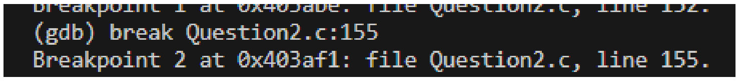
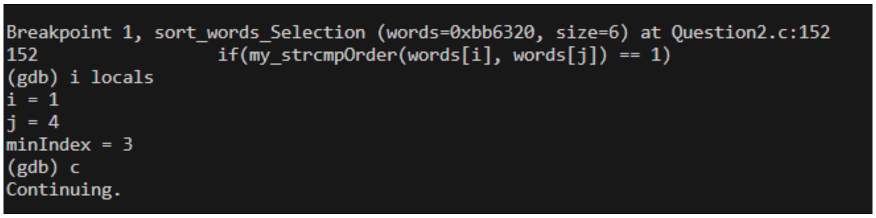
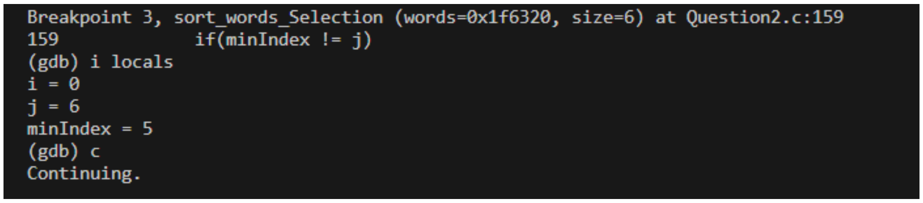
Lab 3 Debugger Report

- Bug Observation 1
  - 

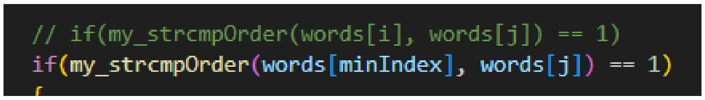
    ```
    There were 3 failures:
    1) TestQ2_readandSort2: testCases.c:362: expected <milan> but was <hello>
    2) TestQ2_readandSort_sel1: testCases.c:383: expected <allison> but was <alice>
    3) TestQ2_readandSort_sel2: testCases.c:401: expected <Physics> but was <McMaster>
    ```
  - Program expected to sort all strings in ASCII order
  - Actual outcome
    - Bubble sort algorithm failed to correctly sort the string in the correct order
- GDB Analysis
  - Breakpoint placed at line 152
    - 

      ```
      (gdb) break Question2.c:152
      Breakpoint 1 at 0x403abe: file Question2.c, line 152.
      (gdb)
      ```
  - Breakpoint placed at line 155:
    - 

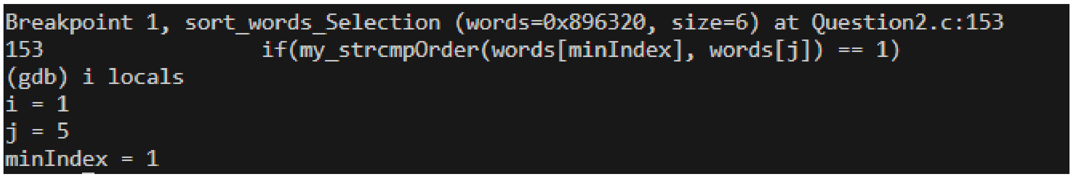
      ```
      Breakpoint 1 at 0x403abe: file Question2.c, line 152.
      (gdb) break Question2.c:155
      Breakpoint 2 at 0x403af1: file Question2.c, line 155.
      ```
  - GDB indicates the program failed at the following
  - Original string: milan, hello, programming, apple, zebra, banana
  - Iteration 1: i = 0, final minIndex = 5 (correct)
    - 

      ```
      Breakpoint 1, sort_words_Selection (words=0xbb6320, size=6) at Question2.c:152
      152                 if(my_strcmpOrder(words[i], words[j]) == 1)
      (gdb) i locals
      i = 1
      j = 4
      minIndex = 3
      (gdb) c
      Continuing.
      ```
      - "apple" is smaller than "milan", minIndex assigned to the index of apple
    - 

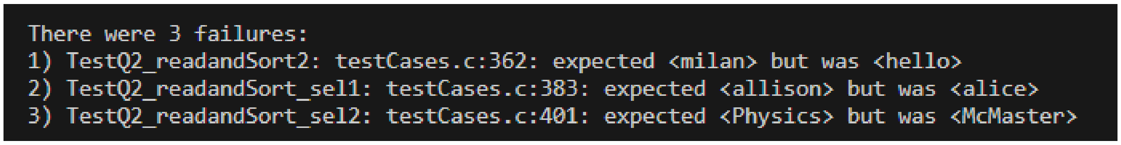
      ```
      Breakpoint 3, sort_words_Selection (words=0x1f6320, size=6) at Question2.c:159
      159             if(minIndex != j)
      (gdb) i locals
      i = 0
      j = 6
      minIndex = 5
      (gdb) c
      Continuing.
      ```
      - However, the program changed the minIndex value to 5 because "banana" is smaller than "milan"
- Possible root cause
  - Incorrect comparison at the if statement
  - Current program compares words[i] and words[j] and reassign minIndex = j when words[i] has a larger ASCII value than words[j]
    - In iteration 2, it was comparing "hello" and "milan", and thought "hello" has a larger ASCII value than "milan".
  - We should be comparing if any of the string besides the starting index has a smaller ASCII value. If it does, it should reassign minIndex with the new smallest value to continue comparing with the remaining list items instead of comparing the remaining list to the starting index.
- Bug Fix Validation

- Line 152 correct to **if(my_strcmpOrder(words[minIndex], words[j]) == 1)**
- 

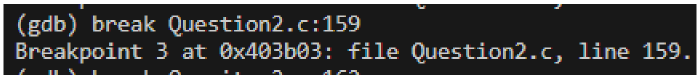
  ```
  // if(my_strcmpOrder(words[i], words[j]) == 1)
  if(my_strcmpOrder(words[minIndex], words[j]) == 1)
  ```

  - Corrected the comparison index, comparing the remain list to the value at the minIndex.
  - New breakpoint at line 153
- Iteration 2: i = 1, final minIndex = 1 (still correct)
  - New string: apple, hello, programming, milan, zebra, banana
  - Incorrect because swapping condition was not implemented correctly
  - 

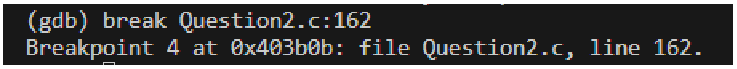
    ```
    Breakpoint 1, sort_words_Selection (words=0x896320, size=6) at Question2.c:153
    153                     if(my_strcmpOrder(words[minIndex], words[j]) == 1)
    (gdb) i locals
    i = 1
    j = 5
    minIndex = 1
    ```

- Bug Observation 2
  - 

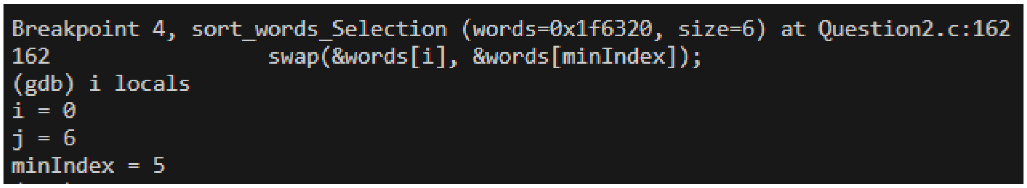
    ```
    There were 3 failures:
    1) TestQ2_readandSort2: testCases.c:362: expected <milan> but was <hello>
    2) TestQ2_readandSort_sel1: testCases.c:383: expected <allison> but was <alice>
    3) TestQ2_readandSort_sel2: testCases.c:401: expected <Physics> but was <McMaster>
    ```

  - Program expected to sort all strings in ASCII order
  - Actual outcome
    - Bubble sort algorithm failed to correctly sort the string in the correct order
- GDB Analysis
  - 

    ```
    (gdb) break Question2.c:159
    Breakpoint 3 at 0x403b03: file Question2.c, line 159.
    ```

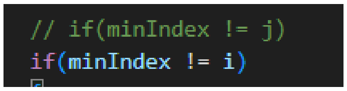
  - Breakpoint at line 159
  - 

    ```
    (gdb) break Question2.c:162
    Breakpoint 4 at 0x403b0b: file Question2.c, line 162.
    ```

  - Breakpoint at line 162
- Possible root cause
  - The comparison of minIndex != j compares minIndex with the last value of j after the iteration which is always going to be 6, which means the swapping will always happen.
  - The comparison disobeys the purpose of selection sort
    - Should be comparing is the minIndex is different than the starting index
    - 

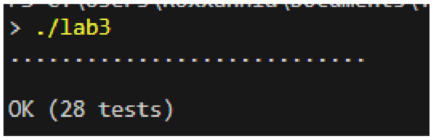
      ```
      Breakpoint 4, sort_words_Selection (words=0x1f6320, size=6) at Question2.c:162
      162                     swap(&words[i], &words[minIndex]);
      (gdb) i locals
      i = 0
      j = 6
      minIndex = 5
      ```

- Bug Fix Validation

- 

  ```
  // if(minIndex != j)
  if(minIndex != i)
  ```

- Program generates correct results, passing all the tests

- 

  ```
  > ./lab3
  ............................

  OK (28 tests)
  ```