

For this program, you will write a series of tests for a linear search function. The search function, some data, and a basic test suite are provided for you, so you will only need to create the code that uses the existing test suite functions to ensure that the recursive factorial function is correct. You should generate at least 10 test cases that cover various possible correct and incorrect inputs for the function. Consider testing cases that the algorithm might have trouble with, such as the first and last elements in the array. The test suite can then be used to ensure that the output of the function matches the expected result.

You should not need to modify the search code or the test suite. You should only need to add the test cases to the test suite file. Since QT SPIM expects all of your code to be in a single file, you can concatenate them together in a few ways. If you are on Windows, you can use the included batch file to do the work for you. If you are having trouble with the batch file, make sure that your file names match those below. You can also use a command line operation.

Windows: copy /Y "Linear Search.asm"+"Test Suite.asm" Output.asm

Unix: cat "Linear Search.asm" "Test Suite.asm" > Output.asm

Your program should include appropriate comments indicating what the code should be doing and what registers are being used for. After displaying the results, your program should exit cleanly. Your programs should be turned in through HuskyCT before class starts on the due date. You should test your programs using the SPIM simulator to ensure their functionality before submitting them.

Sample output:

```
-----Starting functionality tests.-----
Test #1 passed: 1602712035 is in the array.
Test #2 passed: 1041046135 is in the array.
Test #3 passed: -1194480622 is in the array.
Test #4 passed: 736562662 is in the array.
Test #5 passed: -803840256 is in the array.
Test #6 passed: 0 is not in the array.
Test #7 passed: 42 is not in the array.
Test #8 passed: 1492 is not in the array.
Test #9 passed: -96 is not in the array.
Test #10 passed: -8 is not in the array.

-----Starting parameter checking tests.-----
Test #101 passed: Array is null.
Test #102 passed: The array ends outside of a valid part of memory.
Test #103 passed: The array length is less than 1.
Test #104 passed: The array is not word aligned.
-----Testing completed.-----
```

Objectives:

1. To introduce and practice calling functions in the MIPS assembly language.
2. To introduce and practice working with a test suite.