

For this project, you will write an MIPS function to implement the provided C/C++ function. Your code will need to implement the function as written including the recursion and memory access. Your code will not need to generate any new data structures, only change the provided one. Note that this code as written will have numerous side-effects and implementing it properly will reproduce those side-effects.

```
int func (LinkedList* x, int c) {  
    int y;  
    if (x->next != 0) {  
        y = func(x->next, x->data);  
    } else {  
        y = 7;  
    }  
  
    if (x->data == 1 && x->next != 0) {  
        x->data = func(x->next, y);  
    } else if (x->data < 5) {  
        x->data = y * c;  
    } else {  
        x->data *= c;  
    }  
  
    return x->data;  
}
```

The linked list used in the code consists of two elements, a data block and a pointer to a linked list. The data is a single integer and the pointer is the address of the next element in the linked list.

Your program will not need to interact with the console. Instead, you will use a test suite that has been provided for you. The test suite will call your function (func) with the parameters (\$a0 contains the address of a linked list and \$a1 contains a constant.) and will wait for your function to return its results in \$v0. The test suite will also tell you whether your function has correctly calculated the results. Finally, the test suite will also test to make sure that you are following assembly language conventions.

You should attach your code to the test suite for testing by running either:

Windows: copy /Y <Program #4 Function Name>.asm + "Program #4 - Test Suite.asm" <output>.asm

Unix: cat <Program #4 Function Name>.asm "Program #4 - Test Suite.asm" > <output>.asm

The result of either of these commands is a file that contains both sets of code, so that SPIM can load all of it at once. Your final submission should only include your function and not the test suite.

Your program should include appropriate comments indicating what the code should be doing and what registers are being used for. Please include your name and CLID in the program headers and include your CLID in the file names. Your programs should be turned in through Moodle

before class starts on the due date. You should test your programs using the QT-SPIM simulator to ensure their functionality before submitting them.

**Expected output:**

Test #1 passed.
Test #2 passed.
Test #3 passed.
Test #4 passed.
Test #5 passed.
Test #6 passed.
Test #7 passed.
Test #8 passed.
Test #9 passed.
Test #10 passed.
Test #11 passed.
Test #12 passed.
Test #13 passed.
Test #14 passed.
Test #15 passed.

**Objectives:**

1. To review and practice building functions in the MIPS assembly language.
2. To review and practice using the stack in the MIPS assembly language.
3. To introduce and practice working with memory in the MIPS assembly language.