

Project 5
COSC 121 Computer Systems Fundamentals
Spring 2019

Due April 2, 2019 (Tuesday)

Project 5 builds on Project 4. Thus, you may re-use (and modify as needed) your code from Project 4.

REQUIREMENTS

- For the code shown below using a syntax similar to C, write LC-3 assembly code. Your code for *main* should first create an activation record for *main* on the stack. Each time the subroutine *myrecursion* is called, your code should include appropriate steps to create the corresponding activation record.
- Note the following line in the code below:

`b = myrecursion (m-1) + 3*m;`

Your assembly code should first compute the value $3*m$ and store it in register R2. Then the code should make a recursive call to perform *myrecursion* (m-1). The value returned by the recursive call should then be added to the value in R2, and then stored in *temp*.

- Implement your code using “caller-save” convention for saving registers, as well as “callee-save” convention. Thus, you will develop two different assembly programs, one corresponding to each of these conventions. You may use the stack to save the necessary registers.

What to submit:

1. Submit your assembly code files (for both versions).
2. When the code executes, at some points during the execution, the stack will contain the maximum number of activation records for this code. Show the state of the stack the **first time** the stack contains the maximum number of items, and show where the registers are saved (as per the convention used) – labels the stack locations with corresponding register names.

If you are not able to submit a screen shot for the stack, you may draw the contents of the stack by hand in your report.

3. **Report:** Include a report that describes which registers are saved by the caller or the callee in each of your program. Also explain why these registers are being saved. Specify the value of *result* in main after the assignment in the last line in main is completed.

Additional assumptions: If additional assumptions are needed (such as where to locate the code in memory), you may make suitable assumptions.

```
void main()
{
int result;

result = myrecursion (2);
}

int myrecursion (int m)
{
int temp;

temp = 0;

if (m>0)
    temp = myrecursion (m-1) + 3*m;

return (temp);
}
```