

CS120 Lab 07 — MIPS Assembly language coding

We'll use SPIM to write some code. The labs from last week should get you started. Below is a little bit of C code; run it with the C compiler, and see what it does. Then... write the equivalent code in MIPS. You can skip the input/output printing stuff — but make the same changes to the array that happens in C.

There are a few things you'll need to figure out, or complete....

- * create the array in the data section of the assembly code
- * use a register to index through things (counting 0 through 5, inclusive)
- * an if/else statement, along with a comparison
- * indexing into an array, getting a value, writing it back....

To complete the lab, you might want to start by just looping through, to see if you can load the values of the array into a register, one at a time. Once you have that working, add in the IF part of the code. Get that working, then add the ELSE part.

Single step through the code with the simulator, to make sure you understand how everything works....

```
#include <stdio.h>

int array[6] = {1, 2, 3, 4, 5, 6};

int main()
{
    int i;
    for (i = 0; i < 6; i = i + 1)
    {
        if (array[i] < 3)
            array[i] = array[i] + 5;
        else
            array[i] = array[i] + 7;
    }

    // Don't do this in MIPS. Just
    // for illustration in C.
    for (i = 0; i < 6; i = i + 1)
    {
        printf("%d\n", array[i]);
    }
}
```

Note that you can either use base+offset, or move a pointer, when accessing the array. Think a little bit about the different options. Can you do it both ways?

When your code completes, your data in QtSpim should look like this:

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 00000006 00000007 0000000a 0000000b .....
[10010010] 0000000c 0000000d 00000000 00000000 .....
[10010020]..[1003ffff] 00000000
```