

# CS61C Fall 2014 Discussion 3

## 1. Translate the following C code into MIPS.

<pre>// Strcpy: // \$s1 -&gt; char s1[] = "Hello!"; // \$s2 -&gt; char *s2 = //      malloc(sizeof(char)*7); int i=0; do {     s2[i] = s1[i];     i++; } while(s1[i] != '\0'); s2[i] = '\0';</pre>	<pre>addiu \$t0, \$0, 0 Loop: addu \$t1, \$s1, \$t0 # s1[i]       addu \$t2, \$s2, \$t0 # s2[i]       lb  \$t3, 0(\$t1) # char is       sb  \$t3, 0(\$t2) # 1 byte!       addiu \$t0, \$t0, 1       addiu \$t1, \$t1, 1       lb  \$t4, 0(\$t1)       bne \$t4, \$0, Loop Done: sb  \$t4, 1(\$t2)</pre>
<pre>// Nth_Fibonacci(n): // \$s0 -&gt; n, \$s1 -&gt; fib // \$t0 -&gt; i, \$t1 -&gt; j // Assume fib, i, j are these values int fib = 1, i = 1, j = 1;  if (n==0) return 0; else if (n==1) return 1; n -= 2; while (n != 0) {     fib = i + j;     j = i;     i = fib;     n--; } return fib;</pre>	<pre>... beq \$s0, \$0, Ret0 addiu \$t2, \$0, 1 beq \$s0, \$t2, Ret1 addiu \$s0, \$s0, -2 Loop: beq \$s0, \$0, RetF       addu \$s1, \$t0, \$t1       addiu \$t0, \$t1, 0       addiu \$t1, \$s1, 0       addiu \$s0, \$s0, -1       j Loop Ret0: addiu \$v0, \$0, 0       j Done Ret1: addiu \$v0, \$0, 1       j Done RetF: addu \$v0, \$0, \$s1 Done: ...</pre>
<pre>// Collatz conjecture // \$s0 -&gt; n unsigned n; L1: if (n % 2) goto L2; goto L3; L2: if (n == 1) goto L4; n = 3 * n + 1; goto L1; L3: n = n &gt;&gt; 1; goto L1; L4: return n;</pre>	<pre>L1: addiu \$t0, \$0, 2     div \$s0, \$t0 # puts (n%2) in \$hi     mfhi \$t0 # sets \$t0 = (n%2)     bne \$t0, \$0, L2     j L3 L2: addiu \$t0, \$0, 1     beq \$s0, \$t0, L4     addiu \$t0, \$0, 3     mul \$s0, \$s0, \$t0     addiu \$s0, \$s0, 1     j L1 L3: srl \$s0, \$s0, 1     j L1 L4: ...</pre>



0 opcode -> R-type: 000000 00000 00000 10001 01000 000011

sra \$s1 \$0 8