CSE 30321 – Computer Architecture I – Fall 2010
Homework 04 – MIPS Procedure Calls – 75 points
Assigned: September 21, 2010 **– Due:** September 28, 2010

Problem 1: (15 points)

Consider the following MIPS assembly code:

```
f:      sub     $s0, $a0, $a3
        sll     $v0, $s0, 0x1
        add     $v0, $a2, $v0
        sub     $v0, $v0, $a1
        jr      $ra
```

Part A (5 points):
This code contains a mistake that violates the MIPS calling convention. What is this mistake and how should it be fixed?

Part B (5 points):
What is the C equivalent of this code? Assume that the function's arguments are named `a`, `b`, `c`, etc. in the C version of the function.

Part C (5 points):
At the point where this function is called, $a0, $a1, $a2, and $a3 have values 1, 100, 1000, and 30 respectively. What is the value returned by this function? If another function g is called from f, assume that the value returned from g is always 500.

Problem 2: (30 points)

Translate the following functions into MIPS assembly. You may assume that they are leaf procedures, and no $sx registers need to be saved to the stack.

Part A (15 points):

```
int find(int a[], int n, in x) {
      int i;
      for(i=0; i!=n; i++) {
            if(a[i] == x) {
                  return i
            }
      }
      return -1;
}
```

Part B (15 points):

```
int count(int a[] int n, int x) {
      int res = 0;
      int i;
      for(i=0; i!=n; i++) {
            if(a[i] == x) {
                  res = res + 1;
            }
      }
      return res;
}
```

Problem 3: (30 points)
For the following 2 problems, refer to a function *f* that calls another function *func*. The code for C function *func* is already compiled in another module using the MIPS calling convention disussed in class (see also Fig. 2.14 of your text). The function declaration for *func* is:

```
      int func(int a, int b);
```

The code for (2 versions) of function *f* is given in Part A and Part B below. For both Parts A and B, translate function *f* into MIPS assembly. Use the MIPS calling convention. If you need to use register $t0 through $t7, use the lower-numbered registers first.

Part A (15 points):

```
     int f(int a, int b, int c) {
         return func(func(a,b), c);
     }
```

Part B (15 points):

```
      int f(int a, int b, int c) {
            return func(a,b) + func(b,c)
      }
```