Lecture 09: In Class Questions

Assume that you have written the following C code:

```
//---------------------------------------------------------------------
int   variable1   = 10;                  // global variable
int   variable2   = 20;                  // global variable
//---------------------------------------------------------------------
int main(void) {
      int i       = 1;                   // assigned to register s0
      int j       = 2;                   // assigned to register s1
      int k       = 3;                   // assigned to register a3
      int m;
      int n;

      m = addFourNumbers(i, j);

      n = i + j;                         // 1 + 2    = 3

      printf("m is %d\n", m);            // printf modifies no registers
      printf("n is %d\n", n);            // printf modifies no registers
      printf("k is %d\n", k);            // printf modifies no registers
}
//---------------------------------------------------------------------
int addFourNumbers(int x, int y) {
      int i;                             // assigned to register s0
      int j;                             // assigned to register s1
      int k;                             // assigned to register s2

      i     = x + y;                     // 1 + 2    = 3
      j     = variable1 + variable2;     // 10 + 20  = 30
      k     = i + j;                     // 3 + 30   = 33

      return k;
}
//---------------------------------------------------------------------
```

The output of the printf statements in main is: m is 33
n is 3
k is 3

Lecture 09: In Class Questions

Assume this program was compiled into MIPS assembly language with the register conventions described on Slide 3 of Lecture 09. Also, note that in the comments of the program, I have indicated that certain variables will be assigned to certain registers when this program is compiled and assembled. Using a callee calling convention, answer the questions below:

QA: Ideally, how many arguments to the function addFourNumbers must be saved on the stack?

0. By default, arguments should be copied into registers.

QB: What (if anything) should the assembly language for main() do right before calling addFourNumbers?

Copy values of s registers into argument registers; save value of k (in $a3) onto the stack

QC: What is the first thing that the assembly language for addFourNumbers should do upon entry into the function call?

Callee save the s registers

QD: What is the value of register number 2 (i.e. 0010_2) after main completes (assuming there were no other function calls, no interrupts, no context switches, etc.)

33. Register 2 = v0. It should *not* have changed.
(different answer if you assume printf returns value)

QF: Does the return address register ($ra) need to be saved on the stack for this program? Justify your answer. (Assume main() does not return).

No – if no other procedures are called.