CSU34021 Tutorial 3

Q1. Translate the following C/C++ code segment into RISC-1 assembly language (hint: generate unoptimised code first with NOPS in the delay slots and then optimise).

```
int g = 4;

int min(int a, int b, int c) {
    int v = a;
    if (b < v)
        v = b;
    if (c < v)
        v = c;
    return v;
}

int p(int i, int j, int, k, int l) {
    return min(min(g, i, j), k, l);
}

int gcd(int a, int b) {
    if (b == 0) {
        return a;
    } else {
        return gcd(b, a % b);          // call an external function mod to evaluate a % b
    }
}
```

Q2. Consider the following function:

```
int ackermann(int x, int y) {
        if (x == 0) {
            return y+1;
        } else if (y == 0) {
            return ackermann(x-1, 1);
        } else {
            return ackermann(x-1, ackermann(x, y-1));
        }
}
```

Determine, by instrumenting the above code, the number of procedure calls, maximum register window depth, the number of register window overflows and the number of register window underflows that would occur during the calculation of ackerman(3,6) given a RISC-I processor with 6, 8 and 16 register sets respectively. You are NOT asked to translate the ackermann function into RISC-1 code.

See http://en.wikipedia.org/wiki/Ackermann function for some background on the Ackermann function.

Q3 Determine how long it takes to calculate ackermann(3, 6) on your computer. Make sure you time the release version of your code. Briefly describe your approach and comment on its accuracy.