**1)** (10 pts) ANL (Algorithm Analysis)

Below is a program which includes a single function call to the function mysqrt. The function mysqrt includes a while loop. Give an estimate as to how many times that while loop will run during the single function call from main. (Note: a portion of credit for this question will be awarded to the theoretical analysis of the code and the rest of the credit will be awarded to applying that analysis in concert with estimation techniques without a calculator.)

```
int main() {
    printf(“%.6f\n”, mysqrt(1000));
    return 0;
}

double mysqrt(double n) {
    double low = 1, high = n;
    if (n < 1) {
        low = n;
        high = 1;
    }
    while (high - low > .000001) {
        double mid = (low+high)/2;
        if (mid*mid < n)
            low = mid;
        else
            high = mid;
    }
}
```

The difference between high and low at the beginning of the function call is 1000 – 1 = 999. For the purposes of our rough analysis, we can simply call this 1000. The while loop repeatedly divides this difference by 2 for each iteration. Thus, after k iterations the difference between high and low is $\frac{1000}{2^k}$. Thus, we want the minimum value of k such that:

$$\frac{1000}{2^k} < 10^{-6}$$

$$\frac{10^3}{10^{-6}} < 2^k$$

$$2^k > 10^9$$
$$k > log_2 10^9$$

This value is irrational, but we can give an approximation for $10^9$ in terms of a power of 2. Roughly speaking $2^{10}$ (1024) is fairly close to 1000 = $10^{3.}$ So, to get a reasonable rough estimate, we can substitute $2^{30}$ for $10^9$, which will lead to the value of **k ~ 30.**

**Grading: Eqn set up and explanation - 5 pts, solving equation - 3 pts, getting approximation - 2 pts**