**1)** (10 pts) DSN (Recursive Coding)

If Annabelle plays a video game for t minutes, she'll earn f(t) points. (The function f is provided below.) f is a non-decreasing function. (This means that if a < b, then f(a) ≤ f(b).) Annabelle wants to earn at least ***target*** number of points, but since she gets in trouble if she plays too much, she would like to play the least number of minutes that allows her to score at least ***target*** number of points. Complete the function below to be ***recursive*** and efficient (max # of recursive calls should be around 30) to solve the problem. You may assume that 0 < target < 60000, thus it's guaranteed that $0 < t < 10^9$ (for the function f given below.)

```
#include <math.h>

int f(int t) {
    return (int)(2*sqrt(t)+log(t));
}

int minPlay(int target) {
    return minPlayRec(target, 0, 1000000000);
}

int minPlayRec(int target, int low, int high) {

    if (low == high) return low;          // Grading: 2 pts

    int mid = (low+high)/2;               // Grading: 1 pt

    int pts = f(mid);                     // Grading: 2 pt

    if (pts < target)                     // Grading: 1 pt
        return minPlayRec(target, mid+1, high);   // Grading: 2 pts

    return minPlayRec(target, low, mid);          // Grading: 2 pts

}
```

**Grading Notes: Many ways to express the ideas above. One pt off per off by one error. Check these by tracing the low = 2, high = 3 case. One pt total off if target is forgotten in both recursive calls.**