

5. Homework

Due **10/27/08** before class

Please refer to the corresponding exercise sections in the textbook (Rosen, 6th edition).

- 4.5 • (4 points) Use the loop invariant $(I) : power = 2^k$ to show that the code below correctly computes 2^n for any $n \geq 0$. First, use induction to show that (I) is indeed a loop invariant, and then draw conclusions for the termination of the while loop.

```
int power_of_two(int n){
    int power=1;
    int k=0;
    while(k<n){
        // (I) power=2^k
        k++;
        power = power*2;
    }
    return power;
}
```

4.3 (page 308)

- (6 points) 8 a,b,c. First write down the first six elements of the sequence, and then try to find a recursive definition. Do not forget the base case.
- (3 points) 12. Use (weak) induction.
- (2 points) 24 a,b.

4.4 (page 321)

- (2 points) 8
 - (2 points) 10
- c) (3 points) Use weak induction on n to prove that `factorial(n)` correctly computes $n!$, when n is a non-negative integer.

```
int factorial(int n){
    if(n==0)
        return 1;
    return factorial(n-1)*n;
}
```