Data Structures HW 5 - Abraham Murciano

1)
```
int factorCount(int n, int i=1, int q=0){
    //If the iteration variable (i) ≥ the lowest quotient so far (q), all factors have been counted
    if (i >= q && q > 0){
        return 0;
    }
    // If i is a factor
    if (n % i ==0){
        q = n / i; //the lowest quotient is n/i
        int numFactors = (i==q ? 1 : 2); //If i==q, theyre the same factor. Otherwise 2 factors
        return numFactors + factorCount(n, i+1, q);
    } else {
        return factorCount(n, i+1, q);
    }
}
```

2)
```
bool palindrome(string s){
    if(s.length() < 2){
        return true;
    }
    if(s.front()== s.back()){
        s.erase(0, 1); //delete first character
        s.pop_back(); //delete last character
        return palindrome (s);
    } else {
        return false;
    }
}
```

3)
```
bool compareStacks(){
    if (stack1.isEmpty() && stack2.isEmpty()){
        return true;
    }
    if (stack1.isEmpty() || stack2.isEmpty()){
        return false;
    }
    int n1 = stack1.pop(), n2 = stack2.pop();
    bool same = (n1 ==n2);
    if (same){
        same = compareStacks();
    }
    stack1.push (n1);
    stack2.push (n2);
    return same;
}
```

4)
```
int lgFloor(int n, int i=0, int twoPowI = 1) {
    if(n == twoPowI){
        return i;
    }
    if(n < twoPowI){
        return i-1;
    }
    return lgFloor(n, i+1, twoPowI * 2);
}
```