CSI32 Lecture 22: recursion with C++

Drill

1. Trace the call of `fib_rec(6)`
2. How many calls to `fibonacci()` function will be made, if it is called on 7, i.e. `fibonacci(7)`?
3.

Review

1. Every recursive definition can have only one base case. True or False?
2. Is it necessary for the recursive call to be on a smaller parameter?
3. How can you see/spot that the function is recursive?
4. What is *infinite recursion*?
5. What does *stack overflow* mean?
6. Review all examples we did in class.

Exercises:

1. Implement recursive power function

```
long double power_rec(double x)
```

using this idea: $x^n = x \cdot x^{n-1}$, when n > 0 and $x^0 = 1$

2. Recall bit strings: they consist only of digits 0 and 1.
Here are some examples: 000, 010, 01110101, empty string is allowed as well.
Think of a recursive definition of a bit string.