– This represents the number of ways to choose a k-element subset from an n-element set.

1. The expression representing this is
2. It always leads to an integer (not obvious)
3. Direct computation of the expression in 1 leads to overflow, because it is a big number which won’t fit in a “int”, even if the result fits in a 32-bit int. (I am guessing, the division decreases the number).
4. If floats are used, we’ll get a floating result, we want an integer.
5. **Write a program to calculate this without overflowing.**

Solution:

1. Since either calculating the numerator and denominator can lead to overflow, we must figure out a way to cancel out **common** factors between **numerators** and **denominators.**
   1. **This approach doesn’t work, although tempting**.
   2. This doesn’t work because we’ll need to figure out factors, which is a challenging problem, all on its own.
2. **Forget the approach to multiple/divide anything. Instead forget about the expression and try to understand what we are really calculating.**
3. **We know we are calculating the binomial coefficient. The binomial coefficient is the number of ways to choose a k-element subset from an n-element set.**
4. We can
   1. Enumerate k-sized subsets
   2. Choices
      1. We choose nth element – And find (k-1)-sized subsets in rest of the (n-1)-sized set
      2. We don’t choose nth element – And find k-sized subset in rest of the (n-1)-sized set