1. Big O notation, is asymptotic notation.

We say f(n) = O(n), if there exist positive constants N and c so that for all n >= N, f(n) <= c\*g(n).

We say f(n) = Ω(n), if there exist positive constants N and c so that for all n >= N, f(n) >= c\*g(n).

We say f(n) = Θ(n), if there exist positive constants N, c1 and c2 so that for all n >= N,

c1\*g(n) <= f(n) <= c2\*g(n)

we say f(n) = o(n), if for any real positive constant c, there exist N > 0 that for all n >= N, f(n) <= c\*g(n)

1. C++ assert, Expression to be evaluated. If this expression evaluates to 0, this causes an assertion failure that terminates the program. You can use it and then the system will give you the feedback of failure if it fails.

**Programming assignment:**

1. Fast algorithm idea: To store all previous Fibonacci numbers.
2. The series of final digit repeats with a cycle length of 60.
3. Euclidean algorithm for computing greatest common divisor,

GCD(1344, 217)

=GCD(217, 42)

=GCD(42, 7)

=GCD(7, 0)

=7

Each step, a = q1 \* b + r1, when we choose b and r1 separately as new a and new b, have to make sure b > r1.

1. It is easy after knowing how to get GCD, just multiple the two quotients after divided by GCD.
2. There is a Pisano period for all Fibonacci numbers when modulo a particular number.
3. Again, use “The series of final digit repeats with a cycle length of 60” and compute each summation of last digit in the cycle for each n.
4. Make use of method in assignment 6.
5. Change a little bit of codes in assignment 6.