Collatz conjecture:

the following process always stops in a finite # of steps.

Start with $n \in \mathbb{Z}^+$.

until $n = 1$, do this:

if n is even, divide by 2. (a)

else multiply by 3 & add 1. (b)

E.g.: $n = 5$.

$$5 \xrightarrow{(b)} 16 \xrightarrow{(a)} 8 \xrightarrow{(a)} 4 \xrightarrow{(a)} 2 \xrightarrow{(a)} 1.$$

"Lucas sequences"

for $P, Q \in \mathbb{Z}$, define a sequence as follows:

$a_0 = 0$.

$a_1 = 1$.

for $n > 1$, $a_n = P a_{n-1} - Q a_{n-2}$

(Note: if $P = 1$ & $Q = -1$, this is just the Fibonacci sequence.)

Outline of the code:

variables: n, P, Q : input

one before : a_{i-1}

two before : a_{i-2}

current : a_i

} invariant!

Now loop from $i = 2 \ldots n$. current will then be a_n.

Start w/ term $i = 2$.

oneBefore : 1
twoBefore : 0
current : P

```
while (i < n) {
    i++;
    // everything is now out of date!
    twoBefore = oneBefore;
    oneBefore = current;
    current = P * oneBefore - Q * twoBefore;
    // invariant is fixed!
}
cout << current << "\n";
```