

Problem 1

The Algorithms

There are three main algorithms at work in this program in order to compute the n^{th} modified Fibonacci number: the main algorithm, a binary addition algorithm, and a binary to decimal conversion algorithm. Below I will go through each one and explain how each works.

The Main Algorithm –

The way this algorithm works is similar to the second algorithm discussed in class that is used to compute the Fibonacci sequence, instead here we add fib_2 three times rather than just once.

```

If n is 0 or 1
    Output n
If n > 1
    Create three numbers: fib, fib-1, fib-2. The first number fib stores the current calculated
    number in the sequence while fib-1 and fib-2 store the last two numbers in the sequence.
    Loop until n <= 1
        Compute fib = fib-1 + 3 * fib-2
        Set fib = fib-1, fib-1 = fib-2, and fib-2 = fib
        Decrement n by 1
    Output the result (fib-1 since we switched the numbers in the last iteration of the loop)

```

Binary Addition Algorithm–

This algorithm is essentially the simple binary addition algorithm discussed in class.

```

Loop through each digit in the binary numbers
    Add the two current digits plus the carry (initially set to 0)
    If the result is 0
        Set the current result digit to 0 and carry to 0
    If the result is 1
        Set the current result digit to 1 and carry to 0
    If the result is 2
        Set the current result digit to 0 and carry to 1
    If the result is 3
        Set the current result digit to 1 and carry to 1
If carry is 1 then append 1 to the end of the result number

```

Binary to Decimal Algorithm–

This algorithm works on the premise that each digit in a binary number has a corresponding decimal representation. For instance the first bit has decimal 1, the second bit decimal 2, third bit decimal 4, fourth bit decimal 8, and so on. What this algorithm does is to step through each digit of the binary number and add to a decimal number the corresponding decimal amount if that binary bit is set.

```

Start with a decimal number powersOfTwo with stored value of 1
Loop through each binary digit
    If the current binary digit is set to 1
        Add powersOfTwo to our result decimal number
    Add powersOfTwo to itself, thus incrementing to  $2^{n+1}$ 

```

How to Compile and Run

In order to use the program that implements the above algorithms please take fib2.cpp, modify N_FIB near the top to the desired n^{th} sequence number, and compile with your favorite c++ compiler. Run the executable and the result will output in a console/terminal. The program will wait for the user to press enter before quitting.

The Results

Below are the results that the above implementation of the algorithms achieved. The first is the 229th and the second is the 273rd in the modified Fibonacci sequence.

fib₂₂₉ =
25040945258297491662139403047811180857517093442861884558151657174158900739616649
089

fib₂₇₃ =
21763175822254570072588379738462784915492843496263008297041264542590164936613006
2252452324314188828