Goldbach’s Other Conjecture

Function is\_prime(n)

**Input**: n = The number to test for primality

**Output**: True if n is prime, False if n is composite

**Variable**: prime\_flag = a flag that is set to true if n is prime and false if n is composite

Set prime\_flag to True, assuming initially that n is prime

For ii an integer from 1 to n

If ( n is a multiple of ii)

Set prime\_flag to false because n is divisible by ii and thus not prime

Return prime\_flag

**Main Code**

**Variable**: N = max number to check for the squared integer in the conjecture

**Variable:** counter\_example = flag that is true if a counterexample is found, and false otherwise

Initialize N to some large integer, like 1000000

For nn an integer from 3 to N

If nn is prime or even then continue to the next iteration of nn

**Variable:** mm is counter representing the integer to be squared in the conjecture

Initialize mm to 1

While nn – 2m\*\*2 is greater than or equal to 0

If nn – 2m\*\*2 is prime, set counter\_example to false

Increment mm by 1

If counter\_example is true, break out of the loop

Print out nn with appropriate descriptive text, since nn is the first counterexample for the conjecture.