The main program is really simple. We’ll let Exceptions do most of the work.

**int** result = 1;

**try**

{

 **int** x = read();

 **int** p = read();

 **int** q = read();

 // Here are all the ways that the contestant could fail.

 // -- Extra stuff after the 3 integers

 // -- The first number not >3, not <=10^9, or not even

 // -- Either of the last two not prime

 // -- The first number not the sum of the last two

 **if**( *sc*.hasNext()

 || x<=3

 || x>1000000000

 || x%2!=0

 || !isPrime(p)

 || !isPrime(q)

 || x!=p+q ) result = 0;

}

**catch**( Exception e )

{

 // We'll let read() detect all other errors.

 result = 0;

}

*ps*.println( result );

The method read() will make sure a token is a proper integer, throwing an Exception if not.

/\*\*

 \* Read a token, confirm it is a legitimate integer.

 \*

 \* **@return** The integer

 \* **@throws** Exception if anything is awry

 \*/

**private** **int** read() **throws** Exception

{

 String s = *sc*.next();

 **if**( s.startsWith( "-" )

 || s.startsWith( "+" )

 || s.startsWith( "0" ) ) **throw** **new** Exception();

 // Integer.parseInt will throw an Exception if s doesn’t parse as an int

 **return** Integer.*parseInt*( s );

}

Finally, we’ve got to check primality.

/\*\*

 \* Checks if a number is prime.

 \*

 \* **@param** x the number

 \* **@return** true, if x is prime

 \*/

**private** **boolean** isPrime( **int** x )

{

 // Can't be prime if it's <2

 **boolean** result = x>=2;

 **if**( result ) **for**( **int** i=2; i\*i<=x; i++ )

 {

 // Try to find a factor. We only need to go as far as sqrt(x)

 **if**( x%i == 0 )

 {

 result = **false**;

 **break**;

 }

 }

 **return** result;

}