

```
public class problem3_largestPrimeFactor {

    public static void main(String[] args) {
        long num = 600851475143L;
        long largestPrimeFactorOfN = findLargestPrimeFactor(num);

        System.out.println(largestPrimeFactorOfN);
    }

    // create a method to find the largest prime factor of a number
    // this method uses prime factorization
    public static long findLargestPrimeFactor(long n) {
        long largestPrimeFactor = 1L;

        // the smallest possible prime factor is 2, so check that first
        while (n % 2 == 0) {
            largestPrimeFactor = 2;
            // divide n by 2 to simplify number
            n /= 2;
        }

        // now we check odd factors starting with 3
        for (long i = 3; i <= Math.sqrt(n); i += 2) {
            while (n % i == 0) {
                largestPrimeFactor = i;
                n /= i;
            }
        }

        // after all of this factorization, we can determine if n is prime
        // if n is greater than 2 and not divisible by any odds greater than 3, it is
        // at this point we have checked all possible factors, so we can return n
        if (n > 2) {
            largestPrimeFactor = n;
        }

        return largestPrimeFactor;
    }
}
```