```
import java.util.PriorityQueue;
import java.util.Queue;

public class Intro {

    public static void main(String[] args) {
        // sum multiples of 3 and 5
        System.out.println(sumTwoMultiples(3, 5, 1000));

        // sum even fib numbers
        System.out.println(sumEvenFib(4000000));

        // largest prime factor
        System.out.println(largestPrimeFactor(600851475143L));

        // largest palindromic product
        System.out.println(largestPalindromicProduct(3));


    }

    static int sumTwoMultiples(int var1, int var2, int bound) {
        int sum = 0;
        for(int i=0; i < bound; i++) {
            if(i % var1 == 0 || i % var2 == 0) {
                sum += i;
            }
        }
        return sum;
    }

    static int sumEvenFib(int bound) {
        Queue<Integer> queue = new PriorityQueue<Integer>();
        queue.add(1);
        queue.add(2);
        int last = 2;
        int sum = 2;
        while(last < bound) {
            last = last + queue.poll();
            queue.add(last);
            if(last%2==0 && last<bound) {
                sum += last;
            }
        }
        return sum;
    }

    static int largestPrimeFactor(long bound) {
        for(int i = (int) Math.sqrt(bound); i > 1; i--) {
            if (bound%i == 0 && isPrime(i)) {
```

```
                return i;
            }
        }
        return 1;
    }

    static boolean isPrime(int num) {
        if(num == 1 || num == 2) {
            return true;
        }
        for(int i=2;i<Math.sqrt(num);i++) {
            if(num%i == 0) {
                return false;
            }
        }
        return true;
    }

    static int largestPalindromicProduct(int digits) {
        int bound = (int) Math.pow(10, digits) - 1;
        int inc = (int) Math.pow(10, digits - 1);
        for (int k = 1; k < 10; k++){
            for (int i = bound - (k-1)*inc; i > bound - k*inc; i--) {
                for (int j = bound - (k-1)*inc; j > bound - k*inc; j--) {
                    if (isPalindrome(i * j)) {
                        System.out.println("factors = " + i + " x " + j);
                        return i * j;
                    }
                }
            }
        }
        return 0;
    }

    static boolean isPalindrome(int num){
        String num_string = Integer.toString(num);
        int len = num_string.length();
        int split = num_string.length() / 2;
        String left = num_string.substring(0, split);
        StringBuilder right = new StringBuilder(num_string.substring(len-split,
len));
        right.reverse();
        return left.equals(right.toString());
    }
}
```