```
public class Main {
    public static void main(String[] args) {

        // Question 1

        int sum = 0;
        for (int i = 0; i < 1000; i++) {
            if (i % 3 == 0 || i % 5 == 0) {
                sum += i;
            }
        }
        System.out.println("Sum of multiples of 3 and 5 less than 1000: " + sum);


        // Question 2

        int previous = 1;
        int n = 2;
        int fibonacciSum = 0;
        while (n <= 4000000) {
        int temp = n;
        if (n % 2 == 0) {
            fibonacciSum += n;
        }
        n = n + previous;
        previous = temp;
        }
        System.out.println("Sum of even-valued fibonacci terms less than 4000000: "
+ fibonacciSum);


        // Question 3 - takes a while to run

        long composite = 600851475143L / 2;
        while (composite > 0) {
            if (600851475143L % composite == 0) {
                if (checkPrime(composite)) {
                    System.out.println("Largest prime factor of 600851475143: " +
composite);
                    break;
                }
            }
            composite -= 2;
        }


        // Question 4

        int largestPalindrome = 0;
        for (int a = 100; a < 1000; a++) {
            for (int b = 100; b < 1000; b++) {
                int c = a * b;
                if (c >= largestPalindrome) {
                    String temp = Integer.toString(c);
                    String palindrome = "";
                    for (int i = temp.length() - 1; i >= 0; i--) {
                        palindrome += temp.charAt(i);
                    }
                    if (palindrome.equals(temp)) {
                        largestPalindrome = c;
                    }
                }
            }
        }
        System.out.println("Largest palindrome that's a product of two 3-digit 
numbers: " + largestPalindrome);


        // Question 5

        boolean notReached = true;
        int numChecked = 0;
        int counter = 0;
        while (notReached) {
            counter++;
            for (int i = 20; i > 0; i--) {
                if (counter % i == 0) {
                    numChecked++;
                }
            }
            if (numChecked == 20) {
                notReached = false;
            }
            else {
                numChecked = 0;
            }
        }
        System.out.println("Smallest number that is divisible by the first 20 
natural numbers: " + counter);


        // Question 6

        int total = 0;
        for (int k = 1; k <= 100; k++) {
            total += k * k;
        }
        int squared = 0;
        for (int l = 0; l <= 100; l++) {
            squared += l;
        }
        squared = squared * squared;
        System.out.println("Difference between the squared sum of the first 100 
natural numbers and the sum of the squares of the first 100 natural numbers: " + 
(squared - total));


        // Question 7

        int numPrimes = 0;
        int prime = 1;
        while (numPrimes < 10001) {
            prime++;
            if (checkPrime(prime)) {
                numPrimes++;
            }
        }
        System.out.println("10001st prime number: " + prime);
        // Question 8

        String series = 
"73167176531330624919225119674426574742355349194934969835203127745063262395783180169848018694788518438586156078911294949545950173795833195285320880551112540698747158523863050715693290963295227443043557668966489504452445231617318564030987111217223831136222989342338030813533627661428280644448664523874930358907296290491560440772390713810515859307960866701724271218839987979087922749219016997208880937766572733300105336788122023542180975125454059475224352584907711670556013604839586446706324415722155397536978179778461740649551492908625693219784686224828397224137565705605749026140797296865241453510047482166370484403199890008895243450658541227588666881164271714799244429282308634656748139191231628245861786645835912456652947654568284891288314260769004224219022671055626321111109370544217506941658960408071984038509624554443629812309878799272442849091888458015616609791913387549920052406368991256071760605886116467109405077541002256983155200055935729725716362695618826704282524836008232575304207529
63450";
        int largestProduct = 0;
        for (int a = 0; a < series.length() - 15; a++) {
            String subset = series.substring(a, a + 14);
            int product = 1;
            for (int c = 0; c < subset.length(); c++) {
                int digit = Integer.parseInt(subset.substring(c, c + 1));
                product = product * digit;
            }
            if (product > largestProduct) {
                largestProduct = product;
            }
        }
        System.out.println("Largest product of 13 consecutive digits of this 
number: " + largestProduct);


        // Question 9

        int product = 0;
        for (int a = 1; a < 1000; a++) {
            for (int b = 1; b < 1000; b++) {
                int c = 1000 - (a + b);
                if (checkPythagorean(a, b, c)) {
                    product = (a * b * c);
                }
            }
        }
        System.out.println("Product of pythagorean triple whose values sum to 1000: 
" + product);



        // Question 10 - takes a little while to run

        int max = 2000000;
        int sumPrimes = 0;
        int primeNumber = 1;
        while (primeNumber < max) {
            primeNumber++;
            if (checkPrime(primeNumber)) {
                sumPrimes += primeNumber;
```

```java
            }
        }
        System.out.println("Sum of primes less than 2000000: " + sumPrimes);


    }
    public static boolean checkPrime (long n) {
        for (long factor = 2; factor < n / 2 + 1; factor++) {
            if (n % factor == 0) {
                return false;
            }
        }
        return true;
    }

    public static boolean checkPythagorean (int a, int b, int c) {
        if (a * a + b * b == c * c) {
            return true;
        }
        return false;
    }
}
```