```java
public class Program1_MultiplesOf3and5 {
    public static void main(String[] args) {
        int count = 0;
        for (int i = 0; i < 1000; i++) {
            if (i%3==0||i%5==0) {
                count+=i;
            }
        }
        System.out.print(count);
    }
}
```

```java
public class Program2_EvenFibonacci {
    public static void main(String[] args) {
        int a = 1;
        int b = 2;
        int total = 0;

        while(b<4000000) {
            if(b%2==0) total+=b;
            int temp = a+b;
            a = b;
            b = temp;
            System.out.print("a="+a+" b="+b+"\n");
        }
        System.out.println(total);
    }
}
```

```java
public class Program3_LargestPrimeFactor {
    public static boolean isPrime(long num) {
        long a = num / 2;
        while (a > 1) {
            long isFactor = num % a;
            if (isFactor == 0L) {
                return false;
            } else
                a = a - 2L;
        }
        return true;
    }

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

        for (long i = 2; i<num/2; i++) {
            if (isPrime(i) && num % i == 0) {
                num = num / i;
            }
        }

        System.out.println(num);
    }
}
```

```java
public class Program4_LargestPalindrome {
    public static String reverseString(String in) {
        String out = "";
        for (int i = in.length(); i > 0; i--) {
            out += in.substring(i - 1, i);
        }
        return out;
    }

    public static void main(String[] args) {
        int greatest = 0;
        for (int i = 1; i ≤ 999; i++) {
            for (int j = 1; j ≤ 999; j++) {
                int numInt = i * j;
                String num = Integer.toString(numInt);

                if (num.equals(reverseString(num))) {
                    if (numInt > greatest) {
                        greatest = numInt;

                    }
                }
            }
        }
        System.out.println(greatest);
    }
}
```

```java
public class Program5_SmallestMultiple {
    public static void main(String[] args) {
        int[] oneTo20 = new int[20];
        for(int i = 0; i < 20; i++) {
            oneTo20[i] = i + 1;
        }
        outer: for(int i = 1; i < 1000000000; i++) {
            for(int j = 0; j < oneTo20.length; j++) {
                if( i % oneTo20[j] != 0 ) {
                    continue outer;
                }
            }
            System.out.println(i);
            break;
        }
    }
}
```

```java
public class Program6_SumSquareDiff {
    public static int[] sums(int[] a) {
        int sum[] = new int[2];
        for(int i = 0; i < a.length; i++) {
            sum[0] += Math.pow(a[i],2);
            sum[1] += a[i];
        }
        sum[1] = (int)Math.pow(sum[1], 2);
        return sum;
    }

    public static int[] createArray(int len) {
        int[] a = new int[len];
        for(int i = 0; i < len; i++) {
            a[i] = i+1;
        }
        return a;
    }

    public static void main(String[] args) {
        int[] num = createArray(100);
        int[] result = sums(num);
        System.out.println(result[1]-result[0]);
    }
}
```

```java
public class Program7_10001Prime {
    public static boolean isPrime(int num) {
        int a = num-1;
        while (a > 1) {
            int isFactor = num % a;
            if (isFactor == 0) {
                return false;
            } else
                a = a - 1;
        }
        return true;
    }

    public static void main(String[] args) {
        int count = 0;
        int num10001 = 0;
        for(int i = 2; i < 1000000; i++) {
            if(isPrime(i)) {
                count++;
                //System.out.println(i);
            }
            if(count==10001) {
                num10001 = i;
                break;
            }
        }
        System.out.println("count="+count+"\nnum is "+num10001);
    }
}
```

```java
import java.util.Arrays;
/*
 * Product: 2091059712
 * Numbers: [9, 7, 8, 1, 7, 9, 7, 7, 8, 4, 6, 1, 7]
 * */
public class Program8_LargestProductInASeries {
    public static void main(String[] args) {
        int len = 13;
        long greatest = 0L;
        long[] digits = new long[len];
        long product = 1L;
        String numString =
"73167176531330624919225119674426574742355349194934969835203127745063262395783180169848018694788518438586156078911294949545950173795833195285320880551112540698747158523863050715693290963295227443043557668966489504452445231617318564030987111217223831136222989342338030813533627661428280644448664523874930358907296290491560440772390713810515859307960866701724271218839987979087922749219016997208880937766572733300105336788122023542180975125454059475224352584907711670556013604839586446706324415722155397536978179778461740649551492908625693219784686224828397224137565705605749026140797296865241453510047482166370484403199890008895243450658541227588666881164271714799244429282308634656748139191231628245861786645835912456652947654568284891288314260769004224219022671055626321111109370544217506941658960408071984038509624554443629812309878799272442849091888458015616609791913387549920052406368991256071760605886116467109405077541002256983155200055935729725716362695618826704282524836008232575304207529634 50";

        long[] num = new long[numString.length()];
        for (int l = 0; l < numString.length(); l++) {
            num[l] = Integer.parseInt((numString).substring(l, l + 1));
        }

        for (int i = 0; i < num.length - len; i++) {
            product = 1;
            long[] temp = new long[len];
            int k = 0;
            for (int j = i; j < i+len; j++) {
                product *= num[j];
                temp[k] = num[j];
                k++;
            }
            if (product > greatest) {
                greatest = product;
                digits = temp;
            }
        }

        System.out.println("\n" + greatest);
        System.out.println(Arrays.toString(digits));
    }
}
```

```java
public class Program9_SpecialPythTriplet {
    public static void main(String[] args) {
        int loop = 426;
        outer: for(int a = 2; a < loop; a++) {
            for(int b = 3; b < loop; b++) {
                for(int c = 4; c < loop; c++) {
                    if(Math.pow(a,2)+Math.pow(b,2)==Math.pow(c,2)) {
                        if(a+b+c==1000) {
                            System.out.println("1000: "+a+" "+b+" "+c+" ");
                            System.out.println(a*b*c);
                            break outer;
                        }
                    }
                }
            }
        }
    }
}
```

```java
import java.math.BigInteger;

public class Programm10_SummationOfPrimes {

    //method from https://www.baeldung.com/java-prime-numbers.
    //(Created my own isPrime method that worked but it was too slow.
    static boolean isPrime(int num) {
        BigInteger bigInt = BigInteger.valueOf(num);
        return bigInt.isProbablePrime(100);
    }

    public static void main(String[] args) {
        int len = 2000000;
        long sum = 0;
        for(int i = 2; i < len; i++) {
            if(isPrime(i)) {
                sum += (long)i;
            }
        }
        System.out.println("Sum= "+sum);

    }

}
```