ARMSTRONG

```
import java.util.*;

import java.util.Scanner;



public class Armstrong {
   public static boolean isArmstrong(int num, int originalNum) {
            if (num < 10) {
           return num == originalNum;
       }


       int lastDigit = num % 10;
       int remainingDigits = num / 10;
       return isArmstrong(remainingDigits, originalNum) && (int)Math.pow(lastDigit,
String.valueOf(originalNum).length()) == lastDigit;
   }

   public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);
       System.out.print("Enter a number: ");
       int number = scanner.nextInt();
       scanner.close();

       if (isArmstrong(number, number)) {
           System.out.println(number + " is an Armstrong number.");
       } else {
           System.out.println(number + " is not an Armstrong number.");
       }
   }
}
```

AMICABLE

```
public class AmicableNumberCheck{


   public static int sumOfDivisors(int n) {
       int sum = 1;
       for (int i = 2; i <= Math.sqrt(n); i++) {
           if (n % i == 0) {
```

```
                sum += i;
                if (n / i != i) {
                    sum += n / i;
                }
            }
        }
        return sum;
    }


    public static boolean isAmicable(int n) {
        int sum1 = sumOfDivisors(n);
        int sum2 = sumOfDivisors(sum1);

        return sum2 == n && n != sum1;
    }

    public static void main(String[] args) {
        int num= 220;
        if (isAmicable(number)) {
            System.out.println(num + " is an amicable");
        } else {
            System.out.println(num + " is not an amicable");
        }
    }
}
```

AUTOMORPHIC

```
import java.util.*;

class Automorphic{
        public static boolean checkAM(int n){
                int square = n*n;
                while(n > 0){
                        if(n%10 != square%10)
```

```
                return false;

            n /= 10;
            square /= 10;
            }
            return true;
        }

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number: ");
        int num = sc.nextInt();

        if(checkAM(num)){
            System.out.println(num + " is an automorphic number.");
            }
            else{
            System.out.println(num + " is an not automorphic number.");
            }
    }
}
```

GCDRECURSIVE

```
public class GCDRecursive{
    public static int gcd(int a, int b){
        if(b==0){
            return a;
        }
        return gcd(b,a%b);
        }

    public static void main(String[] args){
        int num1 = 24;
        int num2 = 36;

        int gcdResult = gcd(num1, num2);

        System.out.println("GCD of " + num1 + " and " + num2 + " is: " + gcdResult);
```

```
        }
    }
```

LCM RECURSIVE

```
public class LCMRecursive{
    public static int gcd(int a, int b){
        if(b==0)
            return a;
        return gcd(b, a%b);
        }

        public static int lcm(int a, int b){
            return Math.abs(a * b)/gcd(a,b);
            }


        public static void main(String args[]){
            int num1 = 12;
            int num2 = 18;

            int lcmResult = lcm(num1, num2);

            System.out.println("LCM of " + num1 + " and " + num2 + " is: " +
lcmResult);
        }
    }
```

PALINDROME

```
import java.util.*;

class Palindrome{

    public static boolean isPal(String str){
```

```
                str = str.replaceAll("[^a-zA-Z0-9]","").toLowerCase();

                int left = 0;
                int right = str.length() - 1;

                while(left<right){
                        if(str.charAt(left) != str.charAt(right)){
                                return false;
                                }
                                left++;
                                right--;
                                }
                                return true;
        }
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                String input = sc.nextLine();
                sc.close();

                System.out.println(isPal(input));
}}
```

PERFECT NUMBER

```
import java.util.*;

class PerfectNumber{
        public static boolean checkPerfect(int num){
                int sum = 1;
                for(int i =2 ; i <= num/2 ; i++){
                        if(num%i==0){
                                sum += i;
                                }
                        }
                        return sum == num;
        }

        public static void main(String args[]){
                Scanner sc= new Scanner(System.in);
                System.out.println("enter a number: ");
                int num = sc.nextInt();
```

```
                if (checkPerfect(num)) {
            System.out.println(num + " is a perfect number.");
        } else {
            System.out.println(num + " is not a perfect number.");
                        }
                }
            }
```

PRIME CHECK

```
import java.util.*;

class PrimeCheck{
        public static boolean isPrime(int num, int i){
                if(num<2)
                        return false;

                if(num==2)
                        return true;

                if(num%i==0)
                        return false;


                if(i*i<=num)
                        return isPrime(num, i+1);

                else
                        return true;

        }


        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                System.out.println("enter a number: ");
                int num = sc.nextInt();

                System.out.println(num + " is : " + isPrime(num,2));
```

```
		}
}
```

PRIME FACTORS

```
import java.util.*;
public class PrimeFactors{


	static void printPrimeFactors(int num){

		if(num%2==0){
			System.out.println(2 + " " );
			num = num/2;
		}

		for(int i = 3 ; i<=Math.sqrt(num) ; i=i+2){
			while(num%i==0){
				System.out.print(i+ " ");
				num /= i;
				}
		}

		if(num>2){
			System.out.print(num);
		}
}
	public static void main(String args[]){
		Scanner sc = new Scanner(System.in);
		System.out.println("enter a number : " );
		int num = sc.nextInt();

		printPrimeFactors(num);

	}
}
```

RAMANUJAM'S NUMBER

```
public class Ramanujam {

    public static void checkR(int n) {
        boolean found = false;
        for (int a = 1; a <= Math.cbrt(n); a++) {
            for (int b = a; b <= Math.cbrt(n); b++) {
                if (a * a * a + b * b * b == n) {
                    System.out.println(n + " is a Ramanujan number:");
                    System.out.println(a + "^3 + " + b + "^3 = " + n);
                    found = true;
                }
            }
        }
        if (!found) {
            System.out.println(n + " is not a Ramanujan number.");
        }
    }

    public static void main(String[] args) {
        int num = 1729;
        checkR(num);
    }
}
```