Java Programs: Number-Based Logic (A to Z)

Armstrong Number

A number is Armstrong if the sum of cubes of its digits equals the number.
Example: 153 -> $1^3 + 5^3 + 3^3 = 153$

Code:

```
public class Armstrong {
    public static void main(String[] args) {
        int n = 153, sum = 0, temp = n;
        while (n > 0) {
            int digit = n % 10;
            sum += digit * digit * digit;
            n = n / 10;
        }
        if (sum == temp)
            System.out.println("Armstrong Number");
        else
            System.out.println("Not Armstrong");
    }
}
```

Palindrome Number

A number is palindrome if it reads same backward.
Example: 121 -> 121

Code:

```
public class PalindromeNumber {
    public static void main(String[] args) {
        int n = 121, rev = 0, temp = n;
        while (n > 0) {
            int digit = n % 10;
            rev = rev * 10 + digit;
            n = n / 10;
        }
        if (rev == temp)
            System.out.println("Palindrome Number");
        else
            System.out.println("Not Palindrome");
    }
}
```

Perfect Number

Sum of proper divisors equals the number.
Example: 28 -> 1 + 2 + 4 + 7 + 14 = 28

Code:

```
public class PerfectNumber {
    public static void main(String[] args) {
        int n = 28, sum = 0;
        for (int i = 1; i < n; i++) {
            if (n % i == 0) sum += i;
        }
        if (sum == n)
            System.out.println("Perfect Number");
        else
            System.out.println("Not Perfect");
    }
}
```

Strong Number

Sum of factorial of digits = number.
Example: 145 -> 1! + 4! + 5! = 145

Code:

```
public class StrongNumber {
    public static int factorial(int n) {
        int f = 1;
        for (int i = 1; i <= n; i++) f *= i;
        return f;
    }
    public static void main(String[] args) {
        int n = 145, sum = 0, temp = n;
        while (n > 0) {
            int digit = n % 10;
            sum += factorial(digit);
            n = n / 10;
        }
        if (sum == temp)
            System.out.println("Strong Number");
        else
            System.out.println("Not Strong");
    }
}
```

Automorphic Number

A number whose square ends with the number itself.
Example: 76 -> 76^2 = 5776 (ends with 76)

Code:

```
public class Automorphic {
    public static void main(String[] args) {
        int n = 76;
        int square = n * n;
        String s1 = String.valueOf(n);
        String s2 = String.valueOf(square);
        if (s2.endsWith(s1))
            System.out.println("Automorphic Number");
        else
            System.out.println("Not Automorphic");
    }
}
```

Spy Number

Sum of digits = Product of digits.
Example: 1124 -> 1+1+2+4 = 8 and 1*1*2*4 = 8

Code:

```
public class SpyNumber {
    public static void main(String[] args) {
        int n = 1124, sum = 0, product = 1;
        while (n > 0) {
            int digit = n % 10;
            sum += digit;
            product *= digit;
            n = n / 10;
        }
        if (sum == product)
            System.out.println("Spy Number");
        else
            System.out.println("Not Spy");
    }
}
```

Neon Number

Square of number -> sum of digits = number.

Example: 9 -> $9^2 = 81$ -> 8 + 1 = 9

Code:

```
public class NeonNumber {
    public static void main(String[] args) {
        int n = 9;
        int square = n * n;
        int sum = 0;
        while (square > 0) {
            sum += square % 10;
            square = square / 10;
        }
        if (sum == n)
            System.out.println("Neon Number");
        else
            System.out.println("Not Neon");
    }
}
```

Disarium Number

Sum of digits powered by their positions = number.

Example: 135 -> $1^1 + 3^2 + 5^3 = 135$

Code:

```
public class DisariumNumber {
    public static void main(String[] args) {
        int n = 135, temp = n;
        String s = String.valueOf(n);
        int sum = 0;
        for (int i = 0; i < s.length(); i++) {
            int digit = s.charAt(i) - '0';
            sum += Math.pow(digit, i + 1);
        }
        if (sum == temp)
            System.out.println("Disarium Number");
        else
            System.out.println("Not Disarium");
    }
}
```

Harshad Number

A number divisible by the sum of its digits.

Example: 18 -> 1 + 8 = 9, and 18 % 9 = 0

Code:

```
public class HarshadNumber {
    public static void main(String[] args) {
        int n = 18;
        int temp = n, sum = 0;
        while (temp > 0) {
            sum += temp % 10;
            temp = temp / 10;
        }
        if (n % sum == 0)
            System.out.println("Harshad Number");
        else
            System.out.println("Not Harsh");
    }
}
```

Buzz Number

Number ends with 7 or divisible by 7.
Example: 27 or 49

Code:

```
public class BuzzNumber {
    public static void main(String[] args) {
        int n = 27;
        if (n % 10 == 7 || n % 7 == 0)
            System.out.println("Buzz Number");
        else
            System.out.println("Not Buzz");
    }
}
```