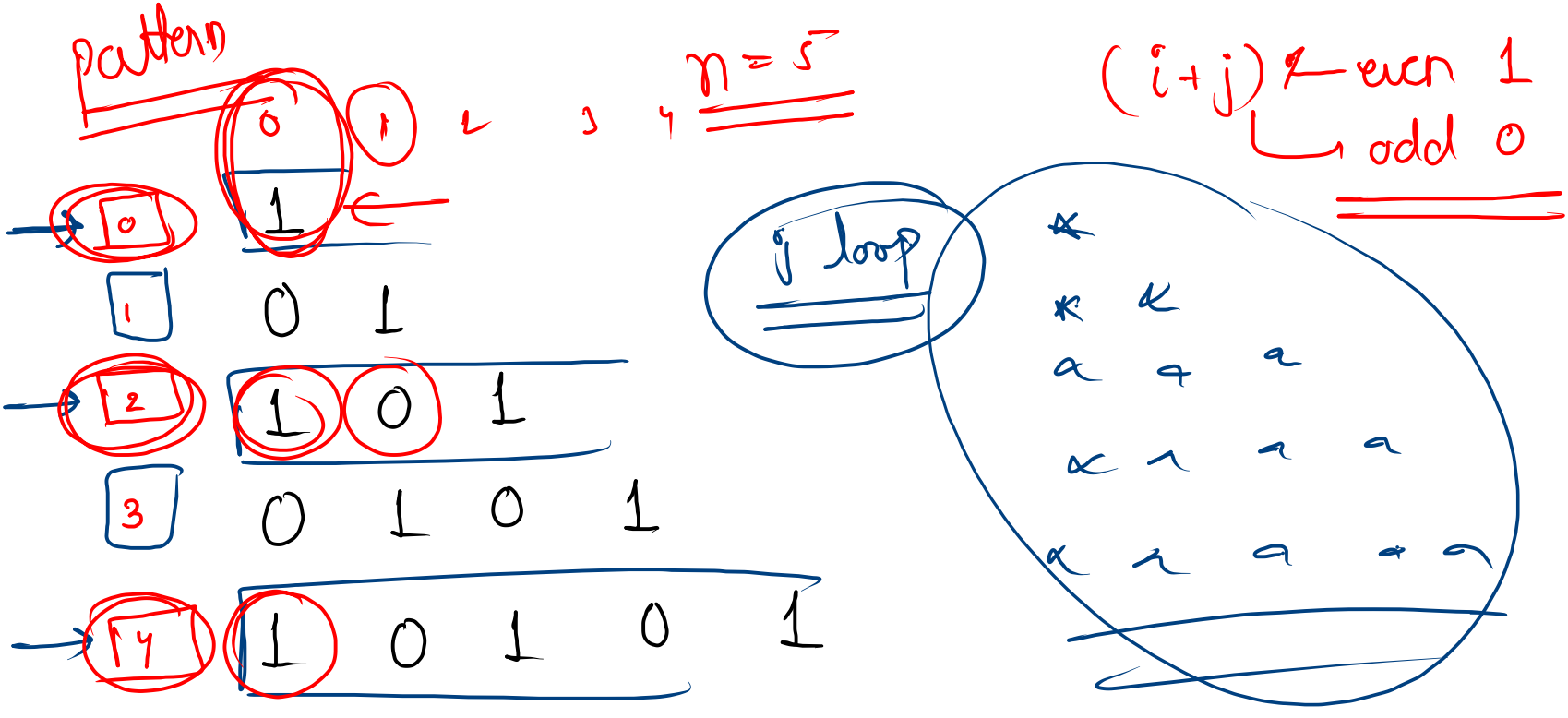

even row ⇐ start = 1 0 1 0 1 0

odd row start = 0 1 0 1 0

Write a function to check if an Armstrong number or not

(note:- input is 3 digit only)

↳ armstrong no. :-

n = 153 digits = 3

ans = $(1)^3 + (5)^3 + (3)^3$

= 1 + 125 + 27

= 153

Yes, armstrong no.

n = 1634, digits = 4

ans = $(1)^4 + (6)^4 + (3)^4 + (4)^4$

= 1 + 1296 + 81 + 256

= 1634

Yes

n = 8

ans = $(8)^1$

= 8

Ex:- n = 153

rem = n % 10 , ans = 0 , n = n / 10 , (until n > 0)

---

rem = 153 % 10 , ans = ans + (3)^3 , n = 153 / 10 , (15 > 0) ✓
= 3 = 15

rem = 15 % 10 , ans = ans + (3)^3 + (5)^3 , n = 15 / 10 , (1 > 0) ✓
= 5 = 1

rem = 1 % 10 , ans = ans + (3)^3 + (5)^3 + (1)^3 , n = 1 / 10 , (0 > 0) ✗
= 1 = 0

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    for (int i = 0; i < n; i++) {
        int num = scn.nextInt();
        boolean ans = checkArmstrongNum(num);
        System.out.println(ans);
    }
}

public static boolean checkArmstrongNum(int n) {
    int temp = n;

    int ans = 0;
    while (n > 0) {
        int rem = n % 10;
        ans = ans + (rem * rem * rem);
        n = n / 10;
    }

    if (ans == temp) {
        return true;
    } else {
        return false;
    }
}
```

n = 123, temp = 123

ans = 0 + (3)^3 + (2)^3 + (1)^3

(123 > 0), rem = 3

(12 > 0), rem = 2

(1 > 0), rem = 1

(0 > 0) ×

if (ans == temp)

(36 == 123)

⇒ Number Theory

Note:-

Math.min(x, y);
Math.max(x, y);

Find GCD 3 / HCF

↳ greatest common divisar

$x = 100, \quad y = 15$

(x % i == 0) && (y % i == 0)

i = 1,	100 % 1 == 0 && 15 % 1 == 0, True	ans = 1
i = 2,	100 % 2 == 0 && 15 % 2 == 0, false	ans = 1
i = 3,	100 % 3 == 0 && 15 % 3 == 0, false	ans = 1
i = 4,	100 % 4 == 0 && 15 % 4 == 0, false	ans = 1
i = 5,	100 % 5 == 0 && 15 % 5 == 0, true	ans = 5
⋮		

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    for (int i = 0; i < n; i++) {
        int x = scn.nextInt();
        int y = scn.nextInt();

        int ans = GCD(x, y);
        System.out.println(ans);
    }
}
```

```java
public static int GCD(int x, int y) {
    int num = Math.min(x, y);
    int ans = 1;
    for (int i = 1; i <= num; i++) {
        if ( x % i == 0 && y % i == 0 ) {
            ans = i;
        }
    }
    return ans;
}
```

Prime checker (no. which can only be divided by 1 or itself)

T = 5

- n = 19, (n%i==0), where i = 2, ..., 18 — Prime
- n = 20, (n%i==0), where i = 2, ..., 19 — i = 2
- n = 93, (n%i==0), where i = 2, ..., 92 — i = 3
- n = 17, (n%i==0), where i = 2, ..., 16 — Prime
- n = 1, (n%i==0), where i = — prime

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int num = scn.nextInt();
        boolean ans = checkPrimeNumber(num);
        if ( ans == true ) {
            System.out.println("Yes");
        } else {
            System.out.println("No");
        }
    }
}
public static boolean checkPrimeNumber(int n) {
    for (int i = 2; i <= n - 1; i++) {
        if ( n % i == 0 ) {
            return false;
        }
    }
    return true;
}
```

Print all factors of a number

n = 20, factors = 1, 2, 4, 5, 10, 20

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    printFactos(n);
}
public static void printFactos(int n) {
    for (int i = 1; i <= n; i++) {
        if ( n % i == 0 ) {
            System.out.println(i);
        }
    }
}
```