

\Rightarrow armstrong no.

\hookrightarrow power is count of digits in no.

\hookrightarrow ans \Rightarrow sum of power of count of digit

$$\overline{xyz^c} \Rightarrow x^c + y^c + z^c \Rightarrow abc$$

check $(abc == xyz) \Rightarrow$ Yes
No

Ques Count no. of digits

num \Rightarrow 1000

C = 4

C = 0

\downarrow
num > 0

Loop

$\left\{ \begin{array}{l} \text{num} = 1000 / 10, \quad C = 1 \\ \text{num} = 100 / 10, \quad \underline{C = 2} \\ \text{num} = 10 / 10, \quad C = 3 \\ \text{num} = 1 / 10, \quad \underline{C = 4} \end{array} \right\}$

$1000 / 10 \Rightarrow \boxed{100} / 10 \Rightarrow 10 / 10 \Rightarrow 1 / 10 \Rightarrow 0$

num \Rightarrow 1000, C = 4

ans $\Rightarrow 1^4 + 0^4 + 0^4 + 0^4$

ans \Rightarrow 1

(ans == num) Yes
No

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

    for (int i = x; i <= y; i++) {
        boolean check = solve(i);
        if (check == true) {
            System.out.println(i);
        }
    }

    public static boolean solve(int num) {
        int count = noOfDigits(num);

        int ans = 0;
        int temp = num;
        while (num > 0) {
            int rem = num % 10;
            ans = ans + (int) Math.pow(rem, count);

            num = num / 10;

            // System.out.println((temp == ans));

            // if (temp == ans) {
            //     System.out.println(true);
            // } else {
            //     System.out.println(false);
            // }

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

            // return (temp == ans);
        }

        public static int noOfDigits(int num) {
            int count = 0;

            while (num > 0) {
                num = num / 10;
                count++;
            }

            return count;
        }
    }
}
```

Def (1)

 $i=2$ $i=3$
$$i = 4$$

A hand-drawn diagram on a white background, enclosed in a red rectangular border. The diagram features a 4x4 grid of cells. The left half of the grid (columns 1 and 2) is filled with red dashed lines. The right half (columns 3 and 4) is filled with yellow solid lines. The numbers 1, 2, 3, 4, and 5 are written in the grid cells. The number 4 is circled in yellow. The numbers are arranged as follows:

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

```
for (int i = 1; i <= n; i++) {
```

```
for(int j=n-1; j>=0; j--) {
    sys0("cc");
}
```

```
for (int j = 1; j <= i; j++) {
    syso(j);
}
```

```
for (int j = 0; j <= 2; j++) {
    Sys0(a[j])
}
```

2

Ques)

input n

↳ n no. of digits input

int n =

Ques)

```
for (int i = 1; i <= n; i++) {  
    if (n % i == 0) {  
        int count = 0;  
        for (int j = 1; j <= i; j++) {  
            if (i % j == 0) {  
                count++;  
            }  
        }  
        if (count == 2) {  
            syso(i);  
        }  
    }  
}
```