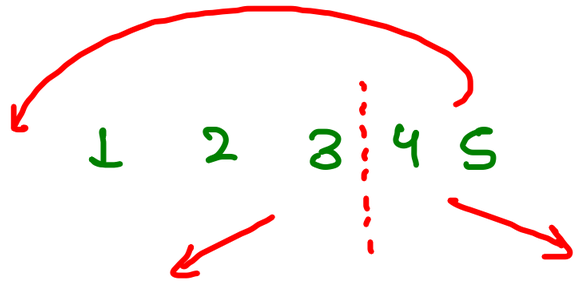
K- Rotate

number.



k = +2 →

4 5 1 2 3

k = +2

+ve -ve

k = 3

num = 1 2 3 4 5

Solution

k = 0 → 1 2 3 4 5

k = 1 → 5 1 2 3 4

k = 2 → 4 5 1 2 3

k = 3 → 3 4 5 1 2

k = 4 → 2 3 4 5 1

k = 5 → 1 2 3 4 5

Khat

Khy

How

k = 0 → 1 2 3 4 5

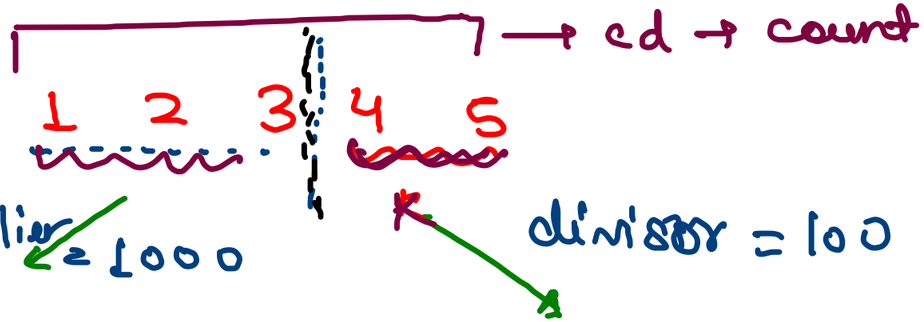
k = -1 → 2 3 4 5 1

k = -2 → 3 4 5 1 2

k = -3 → 4 5 1 2 3

k = -4 → 5 1 2 3 4

num =



45 * 1000 + 123

multiplier

k=2

k=10

num/100 = 123

a

num % 100 → 45

b

(b * Multiplier) + a

✓ cd → count digit

✓ a = num / divisor

✓ b = num % divisor

✓ divisor = 10^k = 1

✓ Multidiv = 10^(cd - k) = 10^5

Result = (b * multiplier) + a

k - Rotated Number

Condition → k +ve. 0 <= k <= c.d.

Conditions for k

num = 1 2 3 4 5 → length = 5

no.

k = 5

Steps -

→ 1. Balance k in Range k = k % length

→ if (k < 0) → (k = k + length)

k = k % length of number

count digit

k = 0 → 1 2 3 4 5

k = -1 → 2 3 4 5 1

k = -2 → 3 4 5 1 2

k = -3 → 4 5 1 2 3

k = -4 → 5 1 2 3 4

k = -5 → 1 2 3 4 5

k = -6 → 2 3 4 5 1

k = -7 → 3 4 5 1 2

k = 0 → 1 2 3 4 5 →

k = 1 → 5 1 2 3 4

k = 2 → 4 5 1 2 3

k = 3 → 3 4 5 1 2

k = 4 → 2 3 4 5 1

k = 5 → 1 2 3 4 5 →

k = 6 → 5 1 2 3 4

k = 7 → 4 5 1 2 3

k = -2

k = k + length

final Step-

- Length of number
- Balance k

mult = 10^(cd-k) loop →
divisor = 10^k →
a = num / ic
b = num % ic
res = [(b * mult) + a]

(i) range k for Range:

k = k % length;

(ii) if (k < 0) {

k = k + length;

}

k is Guaranteed +ve

n = 12

count digit

-12 % 5 =

Math.pow(a,b) → a^b.

double → decim

-a

-ve Remainder

% b := -(a % b)

-12 % 5 = -(12 % 5)

-2

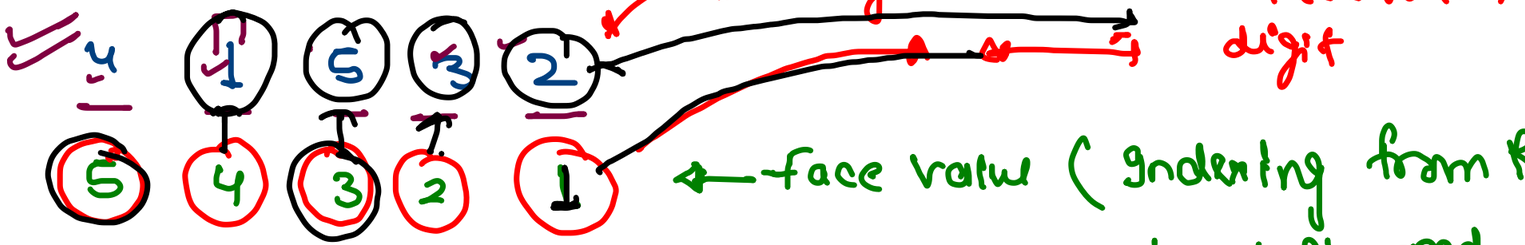
{ -length < k ≤ 0 }

add + length

-length + length < k + length ≤ 0 + length

0 < k + length < length

number →

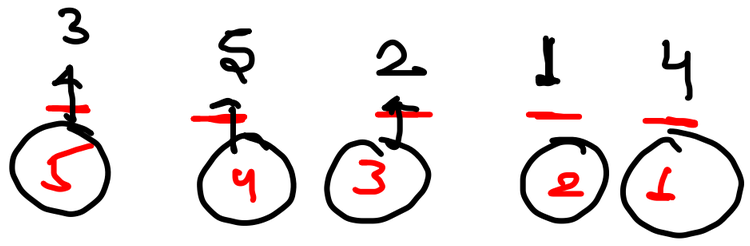


i/p → 41532

digits = { face value }

digits → distinct → digits → 1, 2, 3, 4, 5 → 1 ≤ digit ≤ length & digit

output →



= o/p → 35214.

i/p → 4 1 5 8 2

Steps:

res.

① digit Extraction

② count OR faceValue → res = res + count * 10^(digit-1) ← nth faceval.

digit | faceValue | count | i

digit	faceValue	count	i
2	1	1	0
3	2	2	1
5	3	3	2
1	4	4	3
4	5	5	4

result

1 * 10^1 = 10
2 * 10^2 = 200
3 * 10^4 = 30000
4 * 10^0 = 4
5 * 10^3 = 5000

35214

Sndtr

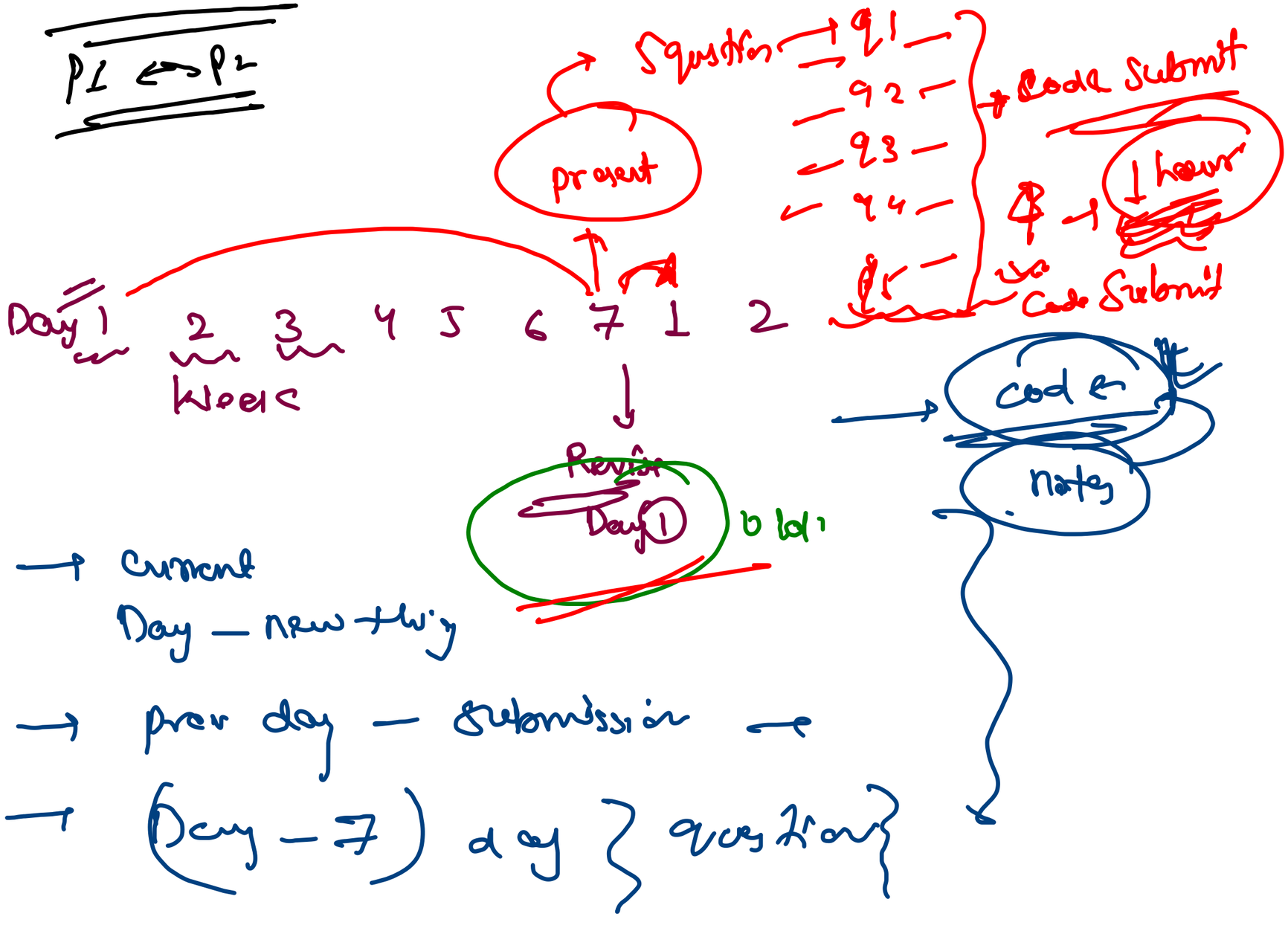
n * 10^(d-1)

35214

output → 35214

sysout(res);

P1 ↔ P2



→ Current Day — new thing

→ prev day — submission →

→ (Day − 7) day } questions

GCD →

a 250, b 30

greatest Number which can divide both exact

division Method

30) 250 (8
240
10) 30 (3
30
0

rem = b % a

b = a;

a = rem;

while (rem != 0)

rem = b

LCM max

LCM (a, b)

$$LCM(a, b) = \frac{a * b}{GCD(a, b)}$$

n1 n2

124, 36

max = 36

36 → n++

lcm % n.

n1 % n == 0 &&

n2 % n == 0

$$LCM(a, b) * GCD(a, b) = a * b$$

6, 12

LCM

a = 1

a = 36

b = 24

2 × 3

2×3

36) 24 (0
0

24) 36 (1
24

24) 36 (1
24

12) 24 (2
24
0

gcd → 12

GCD → 12

24 (2
24
0

$2 \times 2 \times 3$

24 → 2×12

36 → 3×12 $\quad 2 \times 2 \times 3$

LCM

24 12 6 | 2

36 18 9 | 3

2 2 3

$12 \times 3 \times 2 = 72$

$2 \times 3 \times 2$

$8 \times 9 = 72$

Both

Prime factorisation →

num = 1440

Smallest prime no.

div = 2.

2	1440
2	720
2	360
2	180
2	90
2	45
3	15
3	5
5	1

→ 2 2 2 2 2 3 3 5.

if divisible → sysout(div)

reduce no = num / div;

else → div++

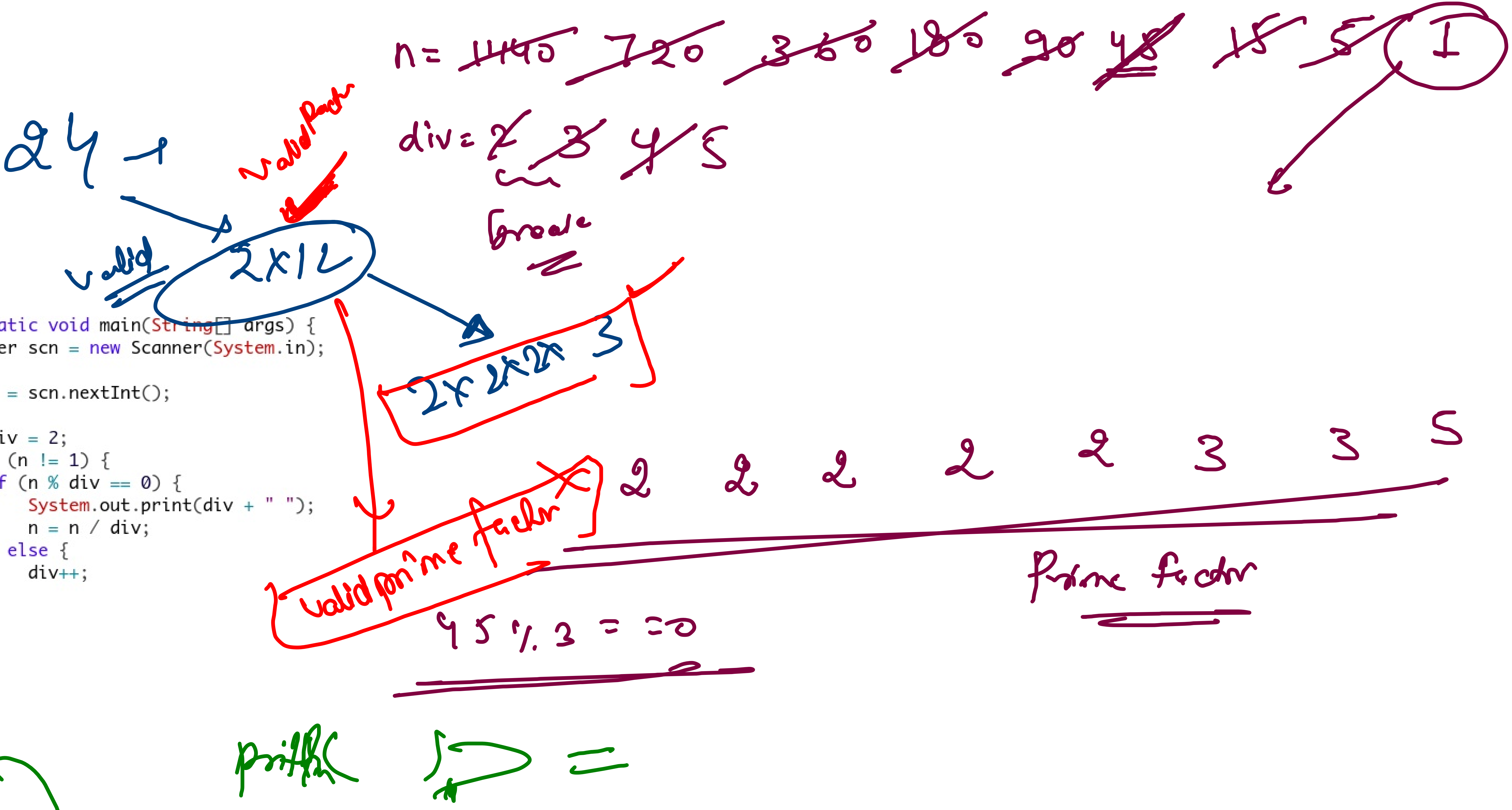
5 → 2 times

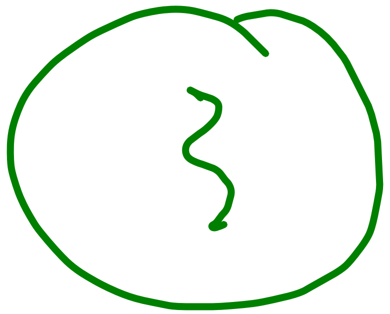
num == 1 → stop break

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

    int n = scn.nextInt();

    int div = 2;
    while (n != 1) {
        if (n % div == 0) {
            System.out.print(div + " ");
            n = n / div;
        } else {
            div++;
        }
    }
}
```





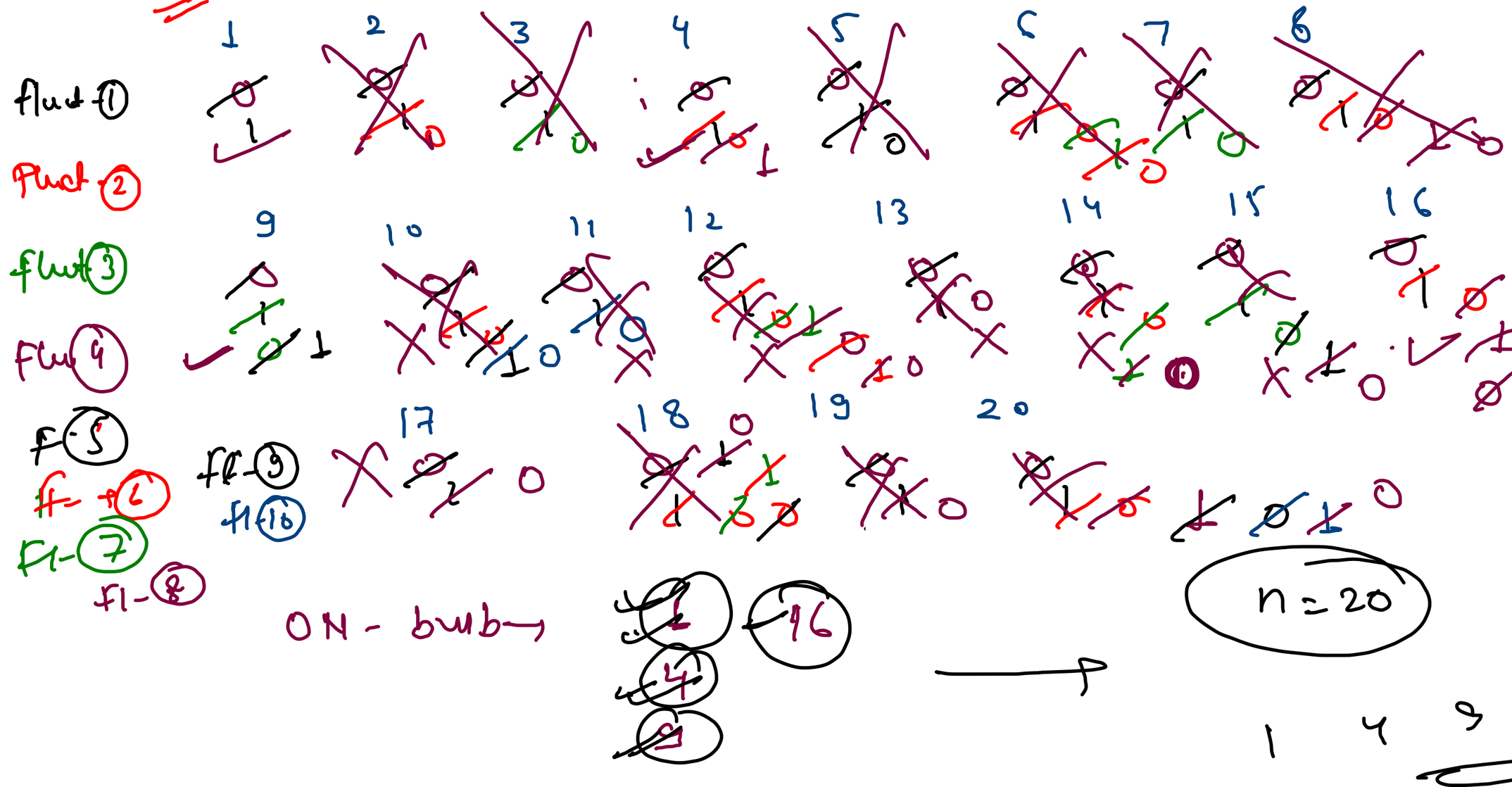
Curious case of benjamin bulb →

i/p → n → nbulbs.

n = 20

① → n times → j fluctuate.

→ Toggle → OFF → ON
ON → OFF



ON - bulb →

1
4
9
16

→

n = 20

1 4 9 16

14 → 1 2 7 14 → ✗ ✗ ✗ ✗ 0

n = 20

1 3 5 7 → ON OFF ON OFF ON

If n is perfect square

1 4 9 16 → ✗ ✗ ✗ ✗ ✓

16 → ✗ ✗ ✗ ✓ ✗

Perfect Square

1 → 20 Perfect Square → odd no factor

1 4 9

1 ≤ i ≤ 4 → i = 1, i = 2, i = 3, i = 4

25 → 1 2 5

Toggle

36 → Repeat

1 × 36, 2 × 18, 3 × 12, 4 × 9

36 × 1, 18 × 2, 12 × 3, 9 × 4

36

6 × 6

Odd no. of factor

38 → 1, 2, 19, 38

1 × 38, 2 × 19

38 × 1, 19 × 2

res = (int) Math.pow(a,b) ⟶ a ^ b. ⟶ o/p → double.

↑ sysout(

int multiplier = (int) Math.pow(10, cd-1); ⟶ Max Range.

int divisor = (int) Math.pow(10, k); ⟶ Decimal Number

$2^n - 1$

res = a^b double res = Math.pow(2,-1): 0.5

2^{-1}

Higher dimension ⟶ lowerdimension] typecasting →

$\frac{1}{2} = 0$

int res = (int) Math.pow(a,b); ⟶ a ^ b.

res = (int) Math.pow(a,b) ⟶ a ^ b. ⟶ o/p → double.

↑ sysout(

int multiplier = (int) Math.pow(10, cd-1); ⟶ Max Range.

int divisor = (int) Math.pow(10, k); ⟶ Decimal Number

$2^n - 1$

res = a^b double res = Math.pow(2,-1): 0.5

2^{-1}

Higher dimension ⟶ lowerdimension] typecasting →

$\frac{1}{2} = 0$

int res = (int) Math.pow(a,b); ⟶ a ^ b.

36 Course

Course flow

38

1x38 38x1

2x19 19x2

36

1x36 → 36x1

2x18 18x2

3x12 12x3

4x9 9x4

6x6

Perfect Square

Group

4 — factors

36 — 0 1 0 1 0 1 0 1 1

doubt support

n=

1 2 | 3 4 5 → n

n-k | k

1000 = divisor

multiplier

mult = $10^{(n-k)}$

divisor = 10^{k}

k = 3

b. 345

a 12

a = num / divisor

b = num % divisor

345 12

345 × 100

. a 12

k = k % length

if (k == 0)

k = k + length

345

12

ans = (b * mult) + a

34500
12

34512

Group