n_1	n_2
18 ,	32

$18 \overline{)32}\ 1$
$\underline{18}$
$14 \overline{)18}\ 1$
$\underline{14}$
$4 \overline{)14}\ 3$
$\underline{12}$
$2 \overline{)4}\ 2$
$\underline{4}$
$\underline{0}$

n_1	n_2
32 ,	18

$32 \overline{)18}\ 0$
$\underline{0}$
$18 \overline{)32}\ 1$
$\underline{18}$
$14 \overline{)18}\ 1$
$\underline{14}$
$4 \overline{)14}\ 3$
$\underline{12}$
$2 \overline{)4}\ 2$
$\underline{4}$
$\underline{0}$

100, 101

$100 \overline{)101}\ 1$
$\underline{100}$
$\boxed{1} \overline{)100}\ 100$
$\underline{100}$
$\underline{0}$

hCD

125, 50

$125 \overline{)50}\ 0$
$\underline{0}$
$50 \overline{)125}\ 2$
$\underline{100}$
$\boxed{25} \overline{)50}\ 2$
$\underline{50}$
$\underline{0}$

hCD

18, 32

$18 \overline{)32}\ 1$
$\underline{18}$
$14 \overline{)18}\ 1$
$\underline{14}$
$4 \overline{)14}\ 3$
$\underline{12}$
$\boxed{2} \overline{)4}\ 2$
$\underline{4}$
$\underline{0}$

hCD

36, 18

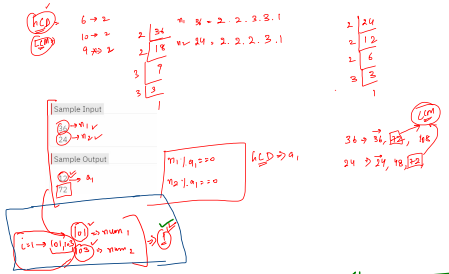
$n_2 \% n_1 == 0 \Rightarrow$ hCD $\Rightarrow n_1$

$36 \overline{)18}\ 0$
$\underline{0}$
$\boxed{18} \overline{)36}\ 2$
$\underline{36}$
$\underline{0}$

36, 24

$36 \overline{)24}\ 0$
$\underline{0}$
$24 \overline{)36}\ 1$
$\underline{24}$
$\boxed{12} \overline{)24}\ 2$
$\underline{24}$
$\underline{0}$

hCD



$\sqrt{101} \Rightarrow \approx 10 \ldots \Rightarrow 10$

$N_1 \times N_2 = LCM \times hCD$ → (1)

hCD → most efficient → Euclidean's theorem → (2)

num1 = ~~36~~ ~~20~~ ~~16~~ 4

num2 = ~~20~~ ~~36~~ ~~20~~ 16

rem = 0

$$N_1 \times N_2 = GCD \times LCM$$

$$\boxed{LCM = \frac{N_1 \times N_2}{GCD}}$$

$$36 \overline{)20} \; 0$$

$$\frac{0}{20 \overline{)36}} \; 1$$

$$\frac{20}{16 \overline{)20}} \; 1$$

$$\frac{16}{4 \overline{)16}} \; 4$$

$$\frac{16}{0}$$

```
// GCD 16 4
while(num2%num1 != 0){
    int rem = num2%num1;   — 1
    num2 = num1;           — 2
    num1 = rem;            — 3
}
```

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~

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

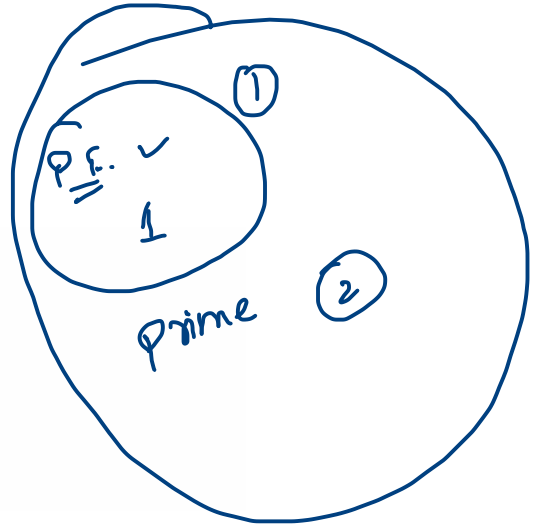
1440 → 2 2 2 2 2 3 3 5

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

	fac	num	fac × fac
✓ I.	2	1440	4
✓ II.	3	45	9
✗ III	4	5	16

2 . 2 . 2 . 2 . 2 . 3 . 3 .

```
int num = scn.nextInt();

for(int fac = 2 ; fac*fac <= num ; fac++){
    while(num%fac == 0){
        num = num / fac;
        System.out.print(fac+" ");
    }
}
```



2	222
3	111
	37

	num	fac	fac x fac
1	222	2	4
2	111	3	9
3	37	4	16
4	37	5	25
2.3 5	37	6	36
6	37	7	49

$\boxed{2.3.37}$

```
int num = scn.nextInt();

for(int fac = 2 ; fac*fac <= num ; fac++){
    while(num%fac == 0){
        num = num / fac;
        System.out.print(fac+" ");
    }
}

if(num != 1){
    System.out.println(num);
}
```

Input

↳ number : no. of bulbs

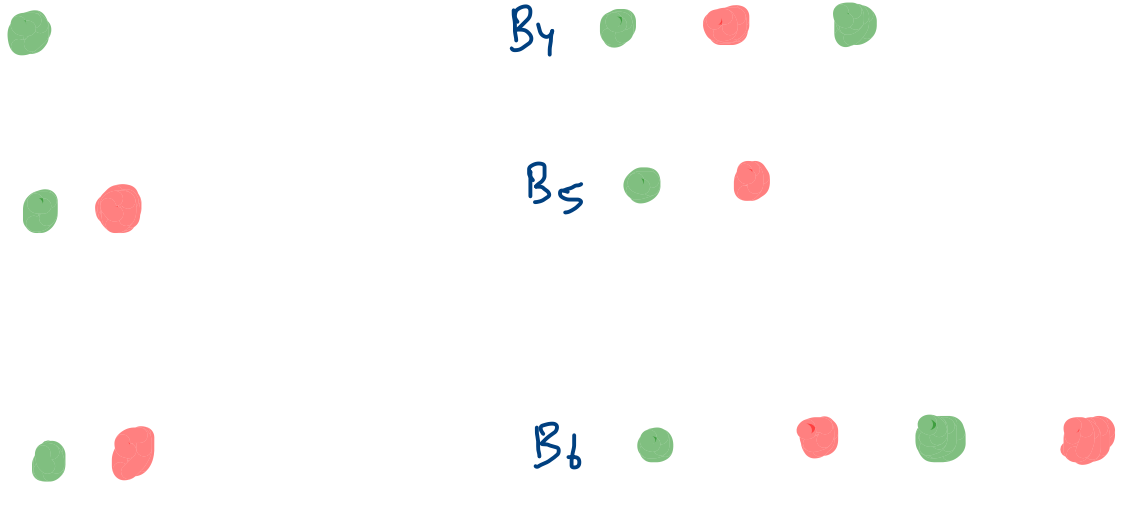
$n = 6$

↳ 6 fluctuations

Initial state = off

on ← Toggle → off

B_1 B_2 B_3 B_4 B_5 B_6



$f = \cancel{1}\ \cancel{2}\ \cancel{3}\ \cancel{4}\ \cancel{5}\ \cancel{6}$

output

1

4

n (Bulbs) ⇒ n fluctuations

fact ⇒ odd no. of toggles → on

$n = 20$

$f = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 17\ 18\ 19\ 20$

on B_1 ✓

off B_2 ✓✓

off B_3 ✓✓

on B_4 ✓✓✓

off B_5 ✓✓

off B_6 ✓✓✓✓

off B_7 ✓✓

off B_8 ✓✓✓✓

on B_9 ✓✓✓

off B_{10} ✓✓✓✓

off B_{11} ✓✓

off B_{12} ✓✓✓✓✓✓

off B_{13} ✓✓

off B_{14} ✓✓✓✓

off ✓ B_{15} ✓✓✓✓

on B_{16} ✓✓✓✓✓

off B_{17} ✓✓

off B_{18} ✓✓✓✓✓✓

off B_{19} ✓✓

off B_{20} ✓✓✓✓✓✓

$1, 4, 9, 16, \ldots \ldots \ldots$ ⇒ perfect square ✓

(1) odd no. of toggle ⇒

(2) perfect square

(3)

$n = 5$

```
for(i=1; i<=n; i++) {
    System.out.print("*");
}
```

```
* * * * *
```

```
for(r=1; r<=n; r++){
    for(i=1; i<=n; i++) {
        System.out.print("*");
    } System.out.println();
}
```

```
1. * * * * *
2  * * * * *
3  * * * * *
4  * * * * *
5  * * * * *
```

$n = 5$

row	nst
1	1
2	2
3	3
4	4
5	5

$\boxed{nst = row}$

6:21 → 6: 26

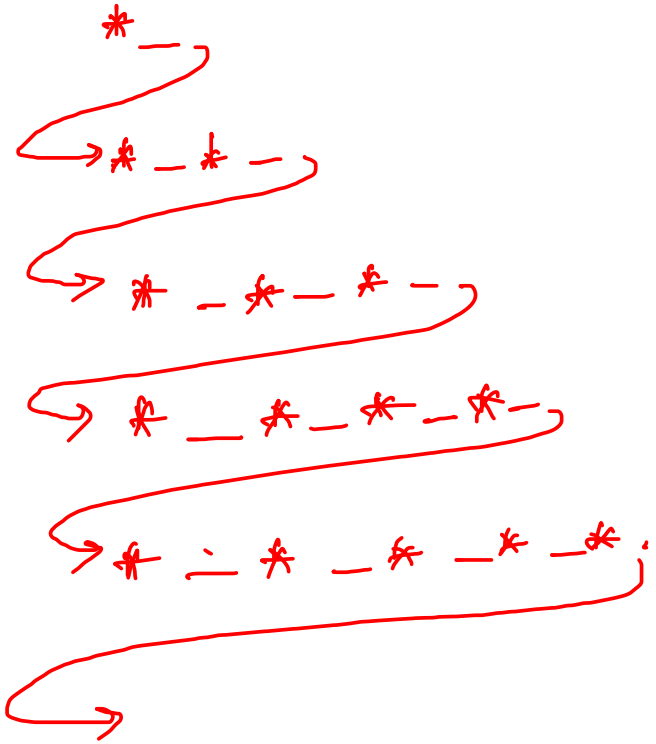
n=5

```java
int n = scn.nextInt();

// code

for(int r = 1 ; r <= n ; r++){
    int nst = r;
     // row : stars print + new line
    for(int i = 1 ; i <= nst ; i++){
        System.out.print("*\t");
    }

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

nst times

r	nst
1	1
2	2
3	3
4	4
5	5

$\boxed{26}$



n = 5

row	nst	row + nst = n+1
1	5	6
2	4	6
3	3	6
4	2	6
5	1	6

row + nst = n+1

nst = n+1 - row

```
for (r=1; r<=n; r++){
    nst = n+1-r;
    for (i=1; i<=nst; i++){
        print("* \t");
    }
    System.out.println();
}
```

Pattern 1

1 × 36	36 × 1
2 × 18	18 × 2
3 × 12	12 × 3
4 × 9	9 × 4

n, n, 36

6 × 6

n + n + 1

total factors ⇒ $\boxed{2 \cdot n + 1}$, n ∈ unique

⇓

odd no.

prime no. → 2 → √n → unique

1 × 18	18 × 1
2 × 9	9 × 2
3 × 6	6 × 3

n, n, 18

Total factors ⇒ $\boxed{2 \cdot n}$, n ∈ unique

⇓

even no.

perfect → odd → on

non-perfect → even → off

```
int n = scn.nextInt();

for(int fac = 1 ; fac * fac <= n ; fac++){
    System.out.println(fac*fac);
}
```

r	nst	nsp
1	1	4
2	2	3
3	3	2
4	4	1
5	5	0
6		

```
_ _ _ _ *
_ _ _ * *
_ _ * * *
_ * * * *
* * * * *
```

n=5

```
1  _ _ _ _ *
2  _ _ _ * *
3  _ _ * * *
4  _ * * * *
5  * * * * *
```

row	nsp	nst
1	4	1
2	3	2
3	2	3
4	1	4
5	0	5

① $\boxed{nst = row}$

② $nsp + nst = n$

$\boxed{nsp = n - nst}$

```
for(r : 1 → n)
    nst ← r
    nsp ← n - nst
    (i = 1 → nsp)
        print("\t")
    (i = 1 → nst)
        print("* \t");
    println();
```

```java
for(int r = 1 ; r <= n ; r++){
    int nst = r;
    int nsp = n - nst;
    // print spaces
    for(int i = 1; i <= nsp ; i++){
        System.out.print("\t");
    }
    // print stars
    for(int i = 1; i <= nst ; i++){
        System.out.print("*\t");
    }
    // new line
    System.out.println();
}
```

26

Wed → 3:30 — 7:00

Pattern 4 → H.W

n=5

→

r	nsp	nst
1	0	5
2	1	4
3	2	3
4	3	2
5	4	1

```java
int n = scn.nextInt();

for (int r = 1; r <= n; r++) {
    int nst = ~~r;~~ n- nsp;
    int nsp = ~~n - nst~~; r-1;
    // print spaces
    for (int i = 1; i <= nsp; i++) {
        System.out.print("\t");
    }
    // print stars
    for (int i = 1; i <= nst; i++) {
        System.out.print("*\t");
    }
    // new line
    System.out.println();
}
```

row { print spaces, print stars }

(i) $\boxed{nsp = r-1}$

(ii) $nsp + nst = n$

$\boxed{nst = n - nsp}$