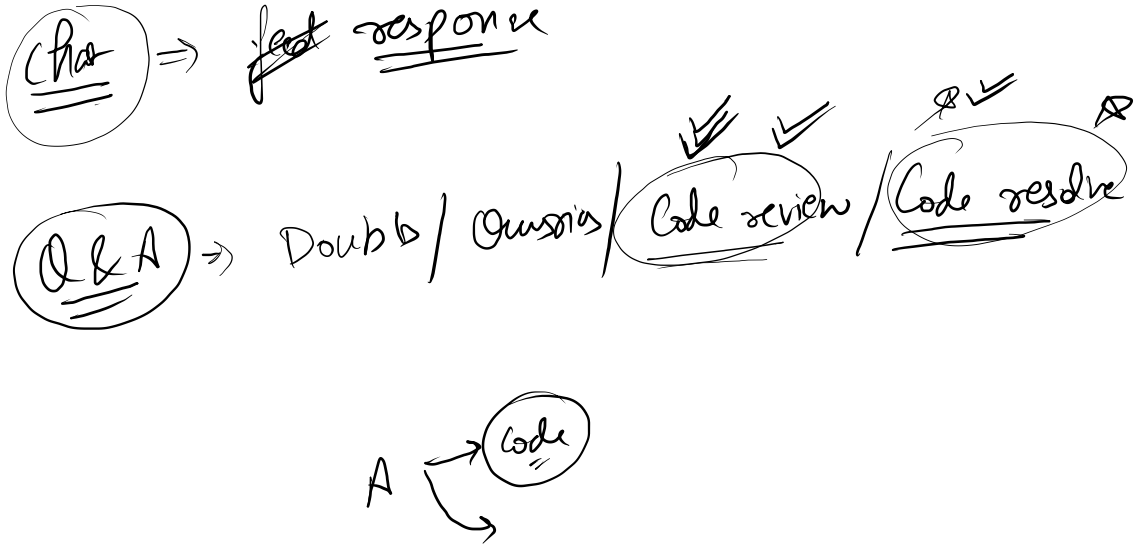
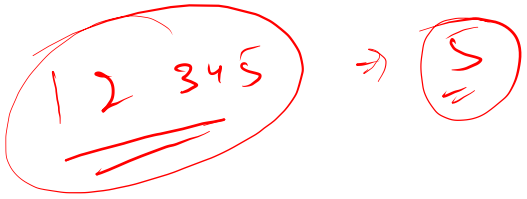
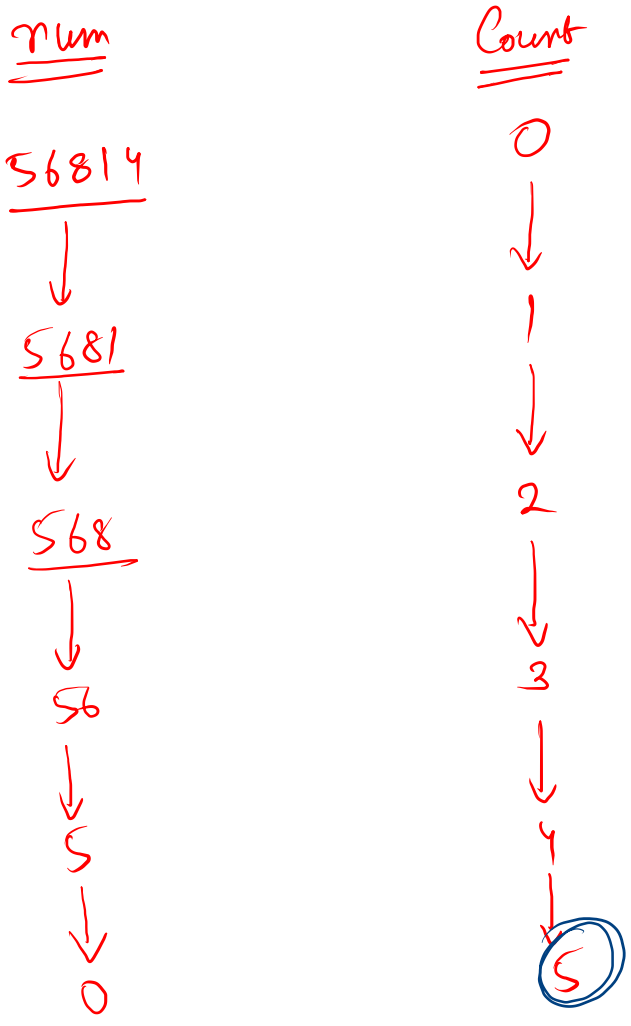

Chat
fast response
Q&A
Doubts
Code review
Code resolve
A
Code

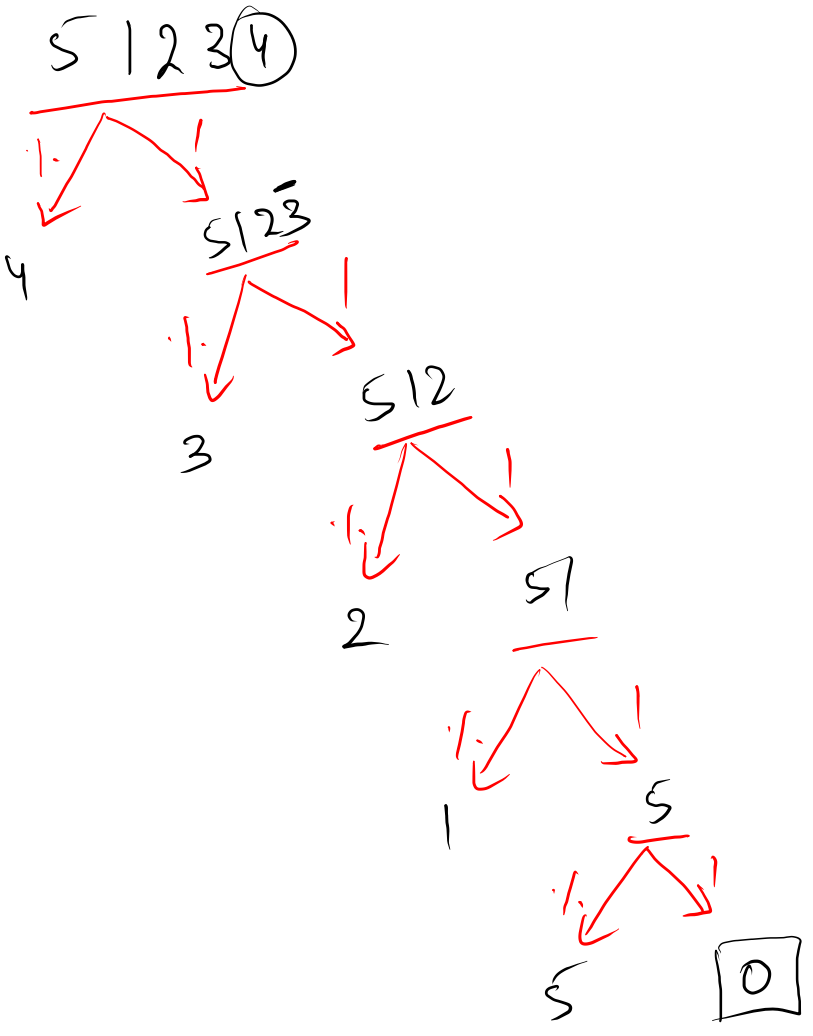

1 2 3 4 5
5

```java
Scanner scn = new Scanner(System.in);

int num = scn.nextInt();

int count = 0;

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

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



H.W-

Reverse a num



num %.10 → digit

```
while(n!=0){
    digit = n %.10;
    n = n/10;
    print(digit);
}
```

28346751

Sample Outpu

73425681

Input

digit = 2 8 3 4 6 7 5 1

pos = 8 7 6 5 4 3 2 1

digit → pos

Output

inv = ~~8~~ 3 4 2 5 6 8 1

pos = 8 7 6 5 4 3 2 1

val(pos) → val(digit)

pos → digit

Inp

inp = 2 8 5 1 4 3 7 6

pos = 9 8 7 6 5 4 3 2 1

= 2 8 3 1 7 5 4 6

pos = 9 8 7 6 5 4 3 2 1

digit → pos

inp = 5 1 2 4 3 = $(5 * 10^4) + (1 * 10^3) + (2 * 10^2) + (4 * 10^1) + (3 * 10^0)$

5 4 3 2 1

→ digit = 4

→ pos = 2

⇒ (digit * 10^(pos-1))

pos → digit

out = 5 2 1 3 4 = $(5 * 10^4) + (2 * 10^3) + (1 * 10^2) + (3 * 10^1) + (4 * 10^0)$

5 4 3 2 1

⇒ (pos * 10^(digit-1))

10^3

$(2 * 10^3)$

$(3 * 10^1)$

x y

Math.pow(x, y);

10

×3 Zeros

basic

Java

pos . $10^{digit-1}$

10 | 5 1 2 4 3

10 | 5 1 2 4

10 | 5 1 2

10 | 5 1

Rem/digits	Pos	
3	1	= $1 \cdot 10^2$ => 100
4	2	= $2 \cdot 10^3$ = 2000
2	3	= $3 \cdot 10^1$ => 30
4		

```
int num = scn.nextInt();
int pos = 1;
int sum = 0;
while(num != 0){
  int digit = num % 10;
  int vl = pos * Math.pow(10,digit-1);
  sum += vl;

  pos++;
  num = num / 10;
}

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

4 bytes 4 8 (double)

sum = 100 + 2000

object = 5kg

box = 10kg

object = 10kg

box = 5kg

Lossy Conversion

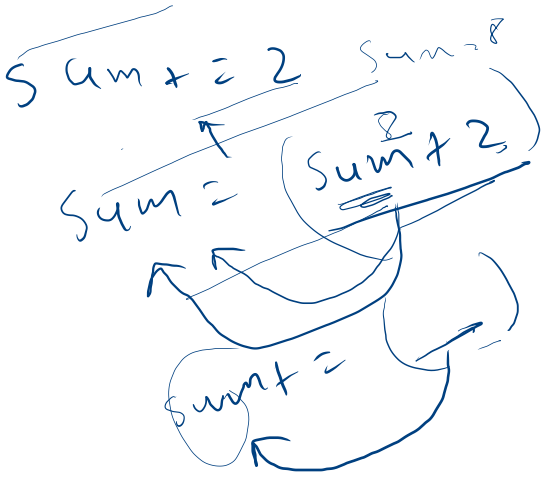
loss hogya

Math.pow()

↳ double => 8 bytes

int => 4 bytes

```
Main.java:14: error: incompatible types: possible lossy conversion from double to int
    int vl = pos *
```


sum += 2
sum += 2
sum += 2
sum +=

$n_1 = 9$ $n_2 = 11$

$9 \overline{)11}\ 1$

$\frac{9}{2} \overline{)9}\ 4$

$\frac{8}{1} \overline{)2}\ 2$

$\frac{2}{0}$

gcd = 1

$(1024, 490)$

$n_1 = 51$ $n_2 = 75$

$51 \overline{)75}\ 1$

$\frac{51}{24} \overline{)51}\ 2$

$\frac{48}{3} \overline{)24}\ 8$

$\frac{24}{0}$

gcd = 3

$n_1 = 6$ $n_2 = 8$

$6 \overline{)8}\ 1$

$\frac{6}{2} \overline{)6}\ 3$

$\frac{6}{0}$

gcd = 2

$n_1 = 12$ $n_2 = 10$

$12 \overline{)10}\ 0$

$\frac{0}{10} \overline{)12}\ 1$

$\frac{10}{2} \overline{)10}\ 5$

$\frac{10}{0}$

gcd = 2

$n_1 = 36$ $n_2 = 24$

$36 \overline{)24}\ 0$

$\frac{0}{24} \overline{)36}\ 1$

$\frac{24}{12} \overline{)24}\ 2$

$\frac{24}{0}$

gcd = 12

$1024 \overline{)490}\ 0$

$\frac{0}{490} \overline{)1024}\ 2$

$\frac{980}{44} \overline{)490}\ 11$

$\frac{484}{6} \overline{)44}\ 7$

$\frac{42}{2} \overline{)6}\ 3$

$\frac{6}{0}$

gcd = 2

HCD & LCM

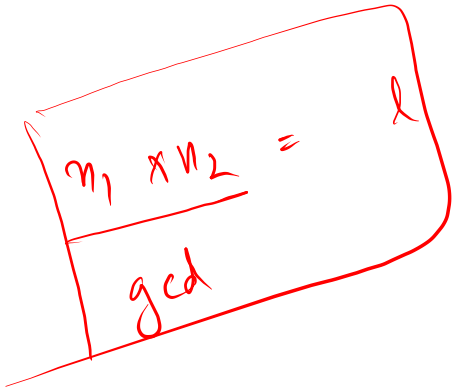
$n_1 = 6$ $n_2 = 8$

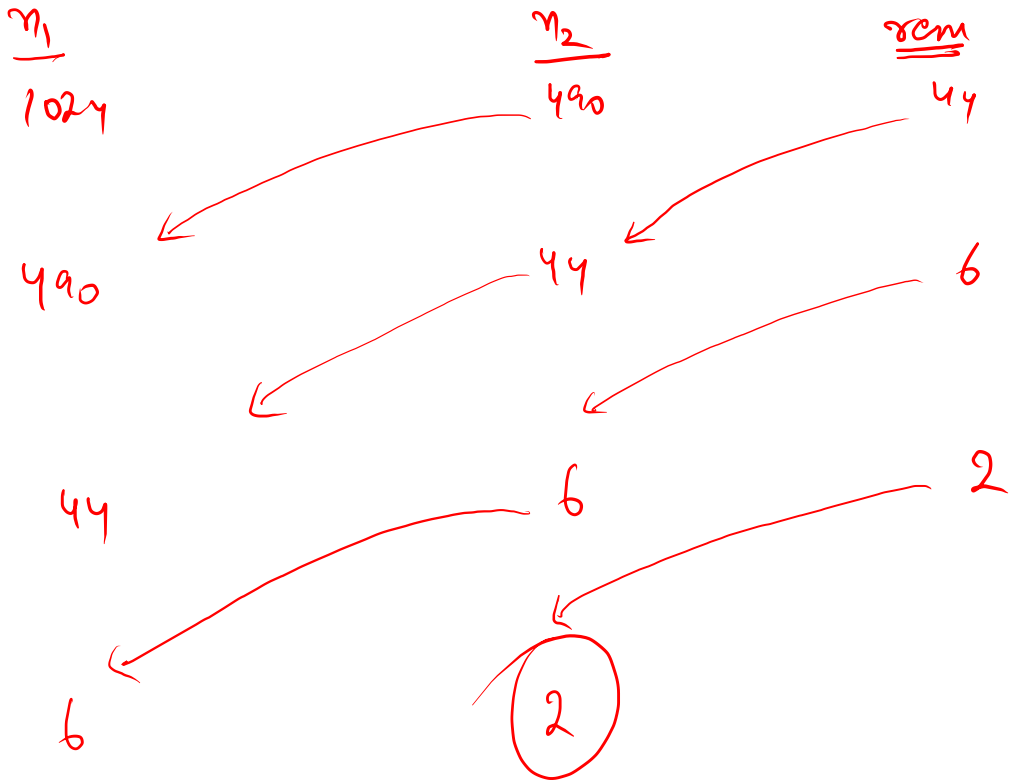
$n_1 \cdot n_2 = \text{gcd} \times \text{lcm}$

check

```
while(n1 % n2 != 0){
    int rem = n1 % n2;
    n1 = n2;
    n2 = rem;
}
```

→1 ✓
→2
→3





$n_1 = 1024$ $n_2 = 490$

$490 \overline{)1024}$

$n_1 = 490$ $n_2 = 1024$

$1024 \overline{)490}\ 0$

0

$490 \overline{)1024}$

Rotation

+ve

-ve

Question

num ⇒ 5 6 2 9 8 (4)

4 5 6 2 9 (8)

8 4 5 6 2 9

Output

k = 2

562984
2

Sample Output

845629

actual rotation
↳ k % no. of digits

(5 6 2 9 8 4) no of digits → 6

+7, +1 = 456298
+8, +2 = 845629
+9, +3 = 984562
+10, +4 = 298456
+5 = 629845
+6 = 562984

+7 = 456298
+8 = 845629
+9 = 984562
+10 = 298456

① after certain rotations num repeats itself
② repetition → no. of digits
③ actual rotation = k % no of digits

+2

K = 50
= (6 * 8) + 2
⇓
original no. +2

k ⇒ 50
= (6 * 8) +2

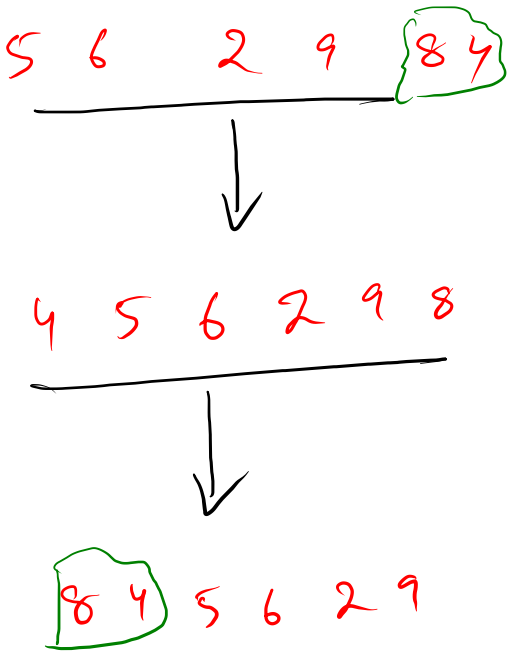
50 = 6 + 6 + 6 + 6 + 6 + 6 + 6 + 6 + 2
original

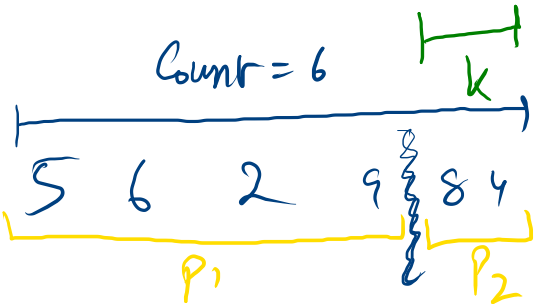
11 → 11 % 6
⇒ 5

5 6 2 9 8 4

1 → 456298
2 → 845629
3 → 984562
4 → 298456
5 → 629845
6 → 562984

7 → 456298
8 → 845629
9 → 984562
10 → 298456
11 → 629845



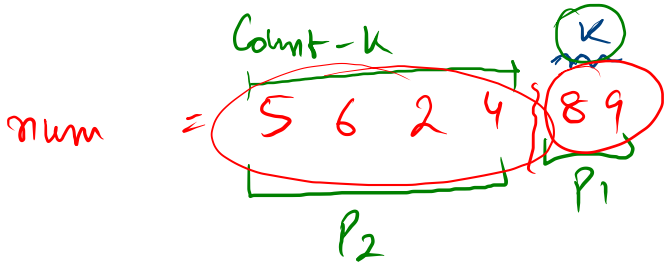


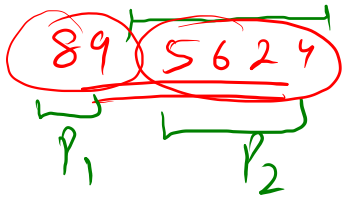
$k = 2$

$\Rightarrow \quad p1 * 10^2 + p2$

$p_2 \rightarrow num \% 10^k$

$p_1 \rightarrow num / 10^k$





tmp1 = 100

P1 = 89

P2 = 5624

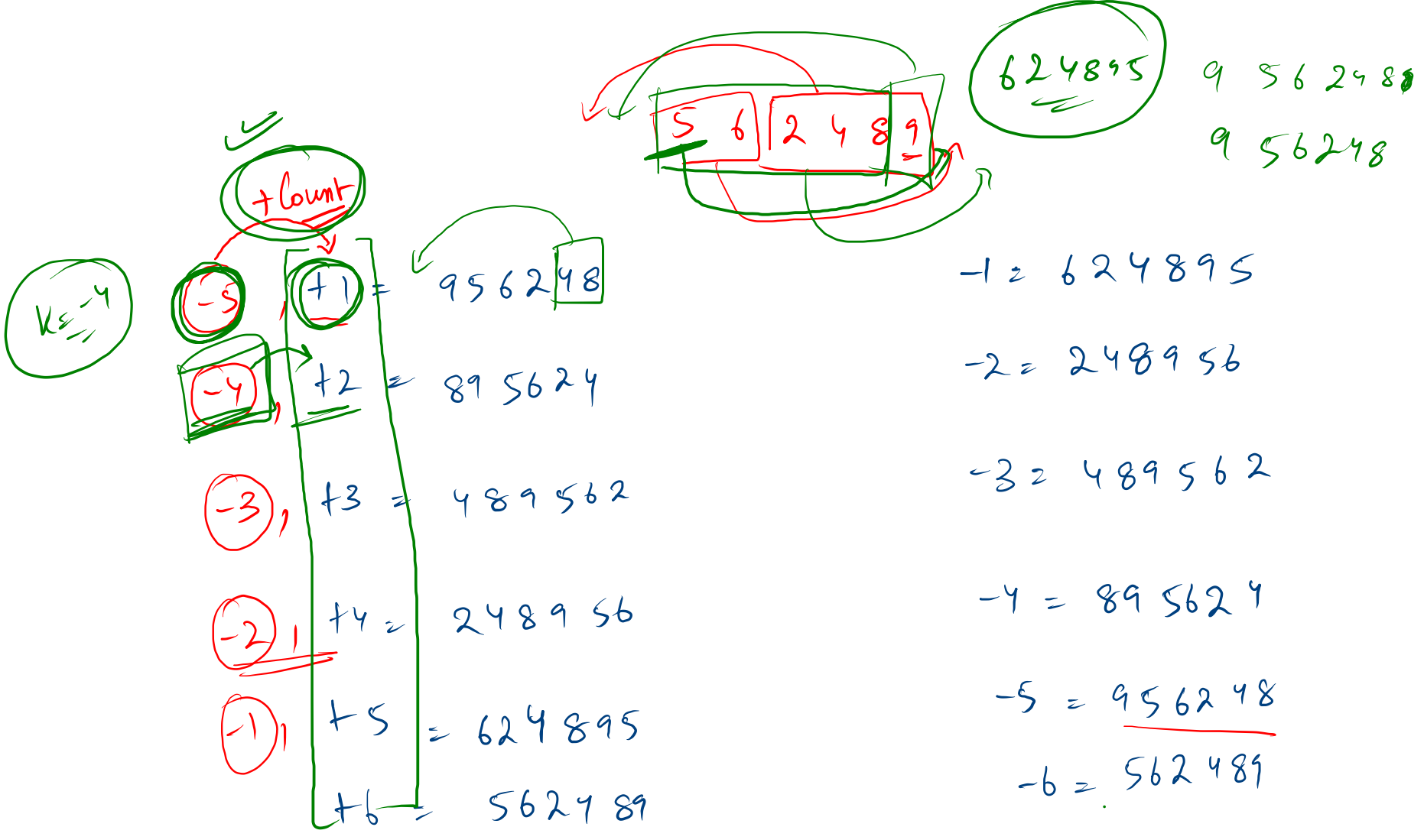
tmp2 = 10^4

= $(89 * 10^4) + 5624$

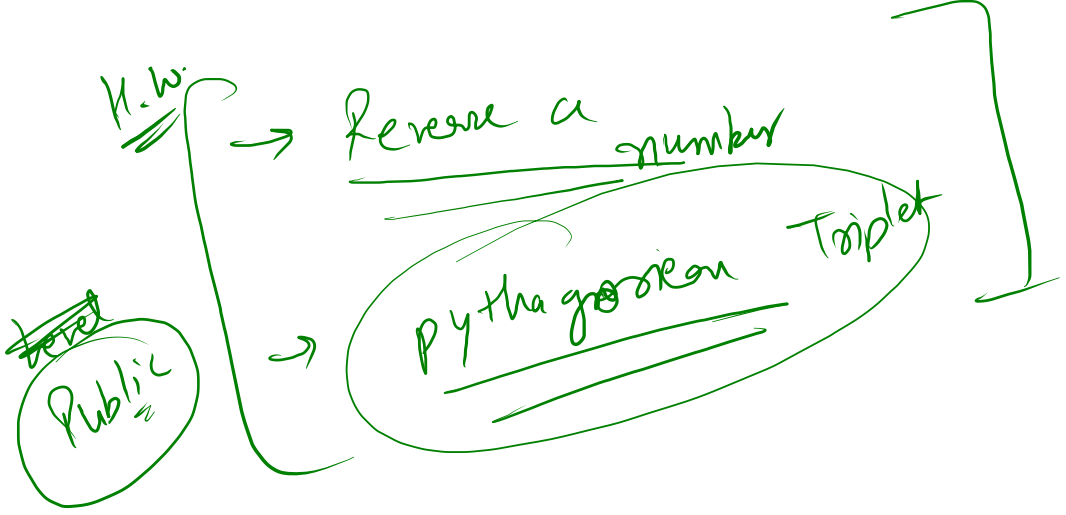
= 890000 + 5624

= 895624

```
int tmp1 = (int)Math.pow(10,k);

int p1 = num % tmp1;
int p2 = num / tmp1;

int tmp2 = (int)Math.pow(10,count-k);

int ans = (p1*tmp2) + p2;
```


624895
9 5 6 2 4 8
9 5 6 2 4 8
5 6 2 4 8 9
+Count
K = -4
-5
+1 = 956248
-4
+2 = 895624
-3
+3 = 489562
-2
+4 = 248956
-1
+5 = 624895
+6 = 562489
-1 = 624895
-2 = 248956
-3 = 489562
-4 = 895624
-5 = 956248
-6 = 562489

```java
int count = 0;
int tnum = num;

while(tnum != 0){
  tnum = tnum / 10;
  count++;
}

// actual rotations
k = k % count;

if(k < 0){ // -ve rotation
  k = k + count;
}

int tmp1 = (int)Math.pow(10,k);

int p1 = num % tmp1;
int p2 = num / tmp1;

int tmp2 = (int)Math.pow(10,count-k);

int ans = (p1*tmp2) + p2;
System.out.println(ans);
```


H.W
Reverse a number
Public
pythagorean Triplet