t = 5

loops → controlling var

j = 1, 2, 3, 4, 5

```
5
13
2
3
4
5
```

Sample Output

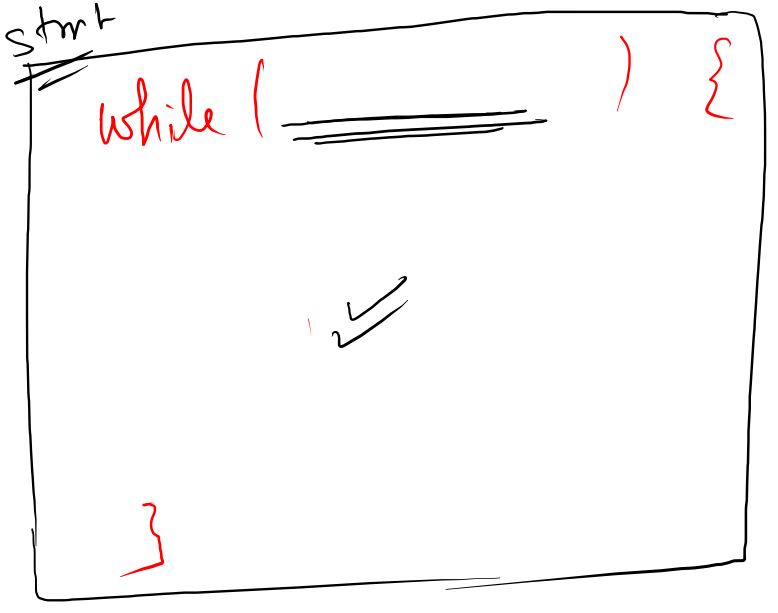
```
prime
prime
prime
not prime
prime
```

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

for(int j = 1 ; j <= t ; j++){
  int num = scn.nextInt();

  int flag = 1; // 1 -> prime
  for(int i = 2 ; i*i <= num ; i++){
      if(num % i == 0){
          // number is not prime
          flag = 0; // 0 -> not prime
          break;
      }
  }

  if(flag == 1){
      System.out.println("prime");
  }else{
      System.out.println("not prime");
  }
}
```

loop

1. initialize, starting point
2. limiting condition / ending condition
3. inc / dec steps

```
for ( initialize ; end cond. ; inc/dec steps ) {

}
```



1 → 10

```
for( int i=1; i<=10 ; i++){
        print(i);
}
```

1 2 3 4 5 6 7 8 9 10

```
int i=1;
while(i<=10){
    print(i);
    i++;
}
```

```
int i = 1;
while(i <= 10){
    System.out.println(i);
    i++;
}
```

1 2 3 4 5 6 7 8 9 10 11

i

1
2
3
4
5
6
7
8
9
10

For ⇄ while

Pattern

```java
int i = 1;
for( ; i <= 10 ; ){
    System.out.println(i);
    i++;
}
```

i

1
2
3
4
5
6
7
8
9
10

```
int i = 1;
for( i++ ; i <= 10 ; i++) {

    Sysprint( i );

}
```

```java
int i = 1;
for(i++ ; i <= 10 ; i++){
    System.out.println(i);
}
```

i = 1 2 3 4 5 6 7 8 9 10 ||

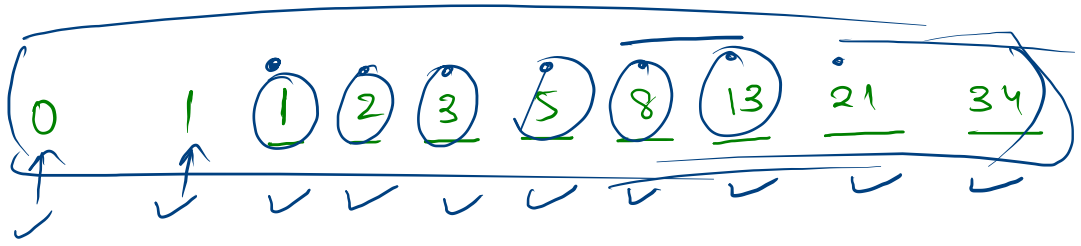

```java
int i = 1;
for( ; i <= 10 ; ){
    System.out.println(i);
    i++;
}
```

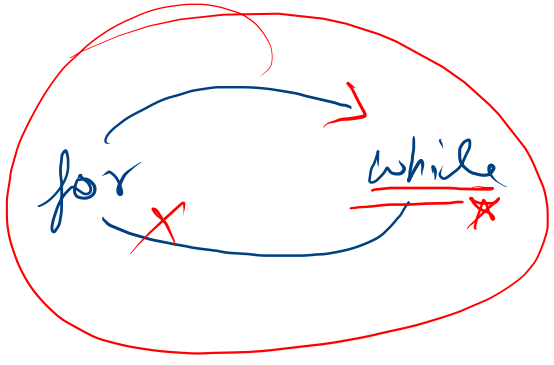
Sample Input

```
10
```

Sample Output

```
0
1
1
2
3
5
8
13
21
34
```





fiboracci Series

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

int first = 0 , second = 1;
int i = 1;
while(i <= n){
  int third = first + second;
  i++;
}
```

loop

n = 5

Dry Run

99 %

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

int n = scn.nextInt();

int first = 0 , second = 1;
int i = 1;
while(i <= n){
  int third = first + second;

  System.out.println(first);

  first = second;
  second = third;
  i++;
}
```

i = 1 2 3 4

i	first	second
1	0	1
2	1	1
3	1	2
4	2	3
5	3	5
6		
7		
8		
9		
10		

first

second

0 1 1 2 3

/ → Integral division

65784383

Sample Output

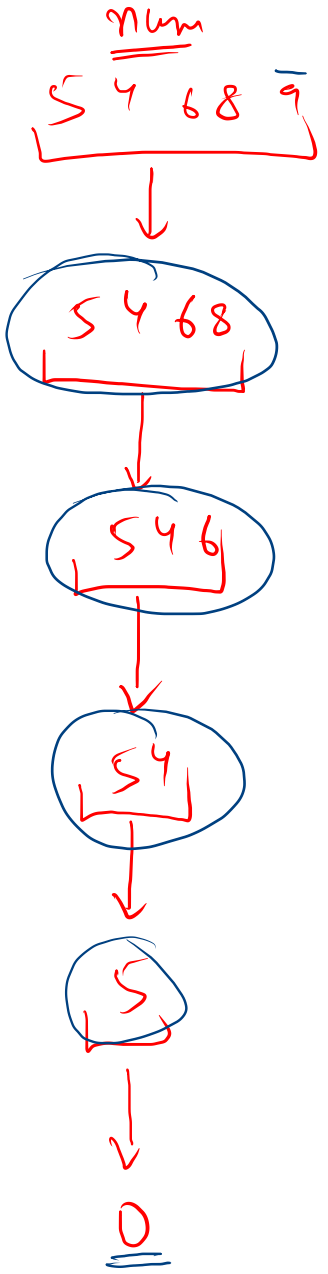
8

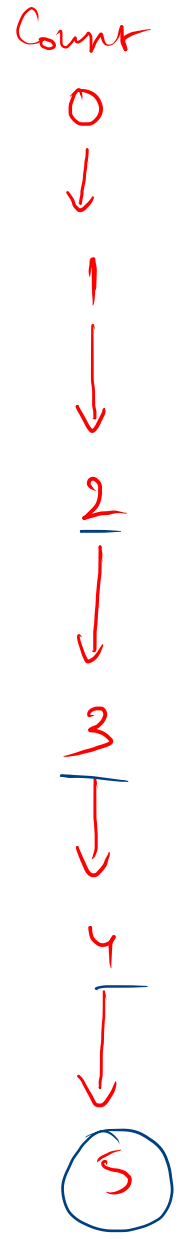
H.W.

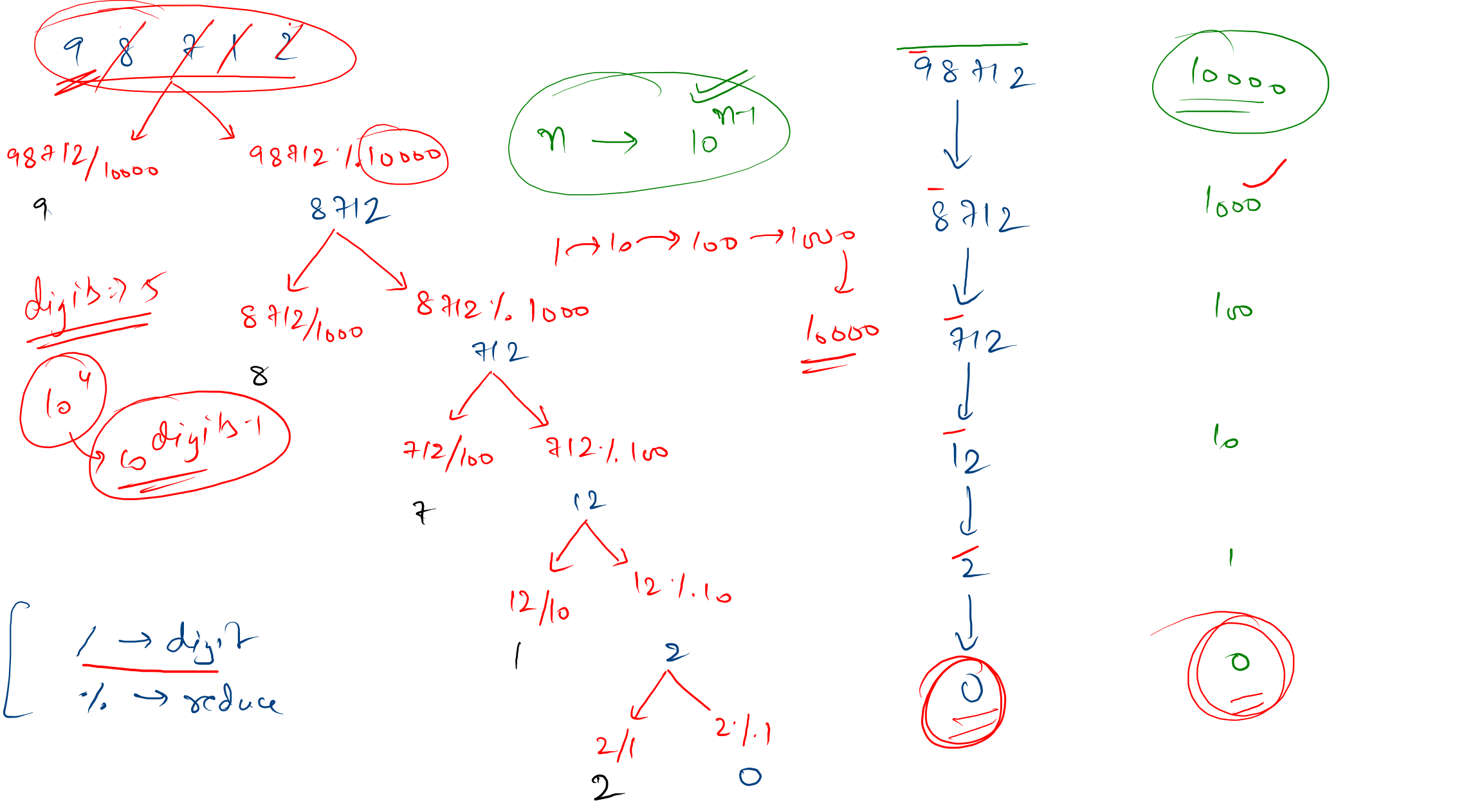
num

1. num = num/10 ;
2. Count++ ;

num != 0






9 8 7 1 2
98712/10000
98712 % 10000
9
8712
8712/1000
8712 % 1000
8
digits = 5
10^4
10 digits-1
n → 10^(n-1)
1 → 10 → 100 → 1000
10000
712
712/100
712 % 100
7
12
12/10
12 % 10
1
2
2/1
2 % 1
2
0
[/ → digit
% → reduce
98712
8712
712
12
2
0
10000
1000
100
10
1
0

```
// tmp
int tmp = 1;
while(n > 9){
   n = n / 10;
   tmp = tmp * 10;
}
```

num

4 8 7 1 2

4 8 7 1

4 8 7

4 8

4

tmp

$1 = 10^0$

10^1

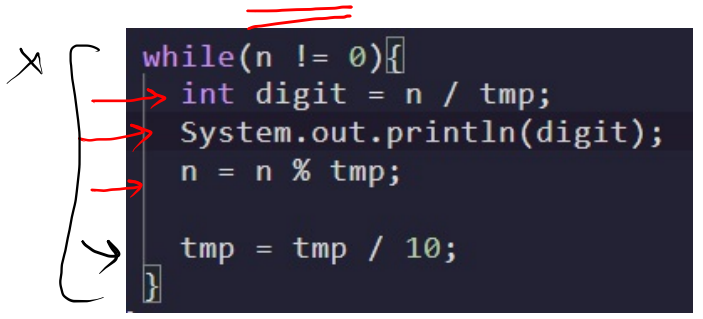
10^2

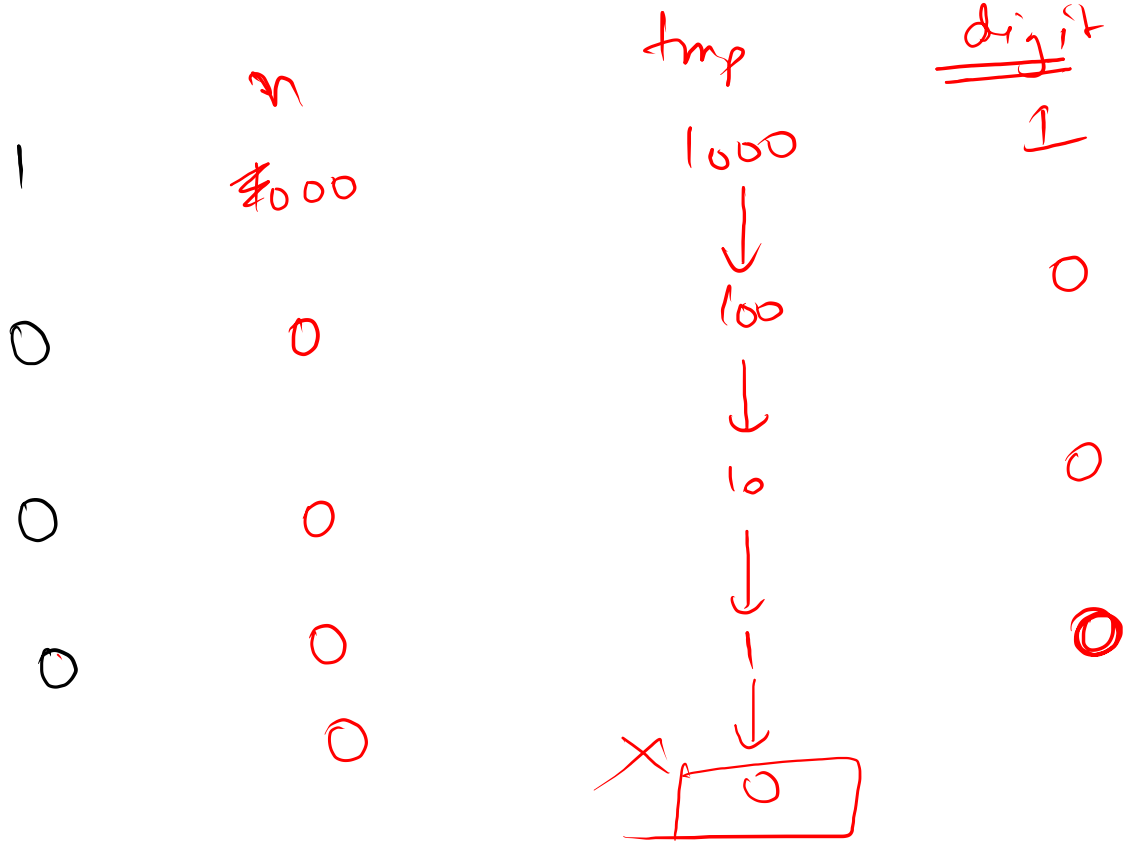
$1000 \approx 10^3$

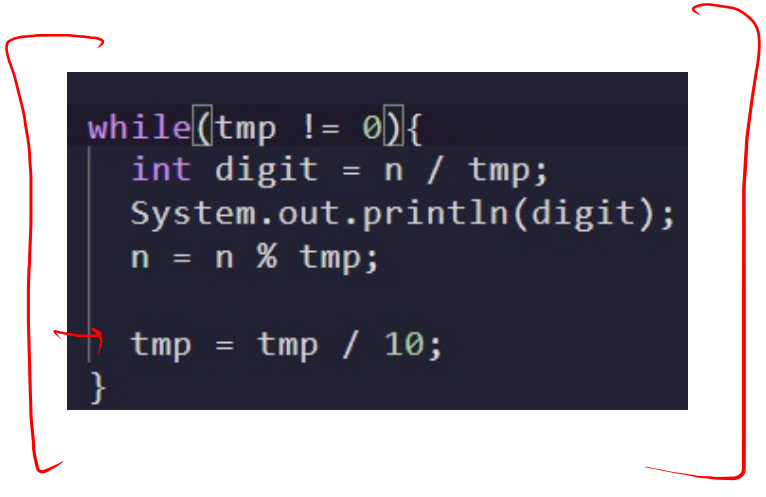
$10000 \approx 10^4$


1

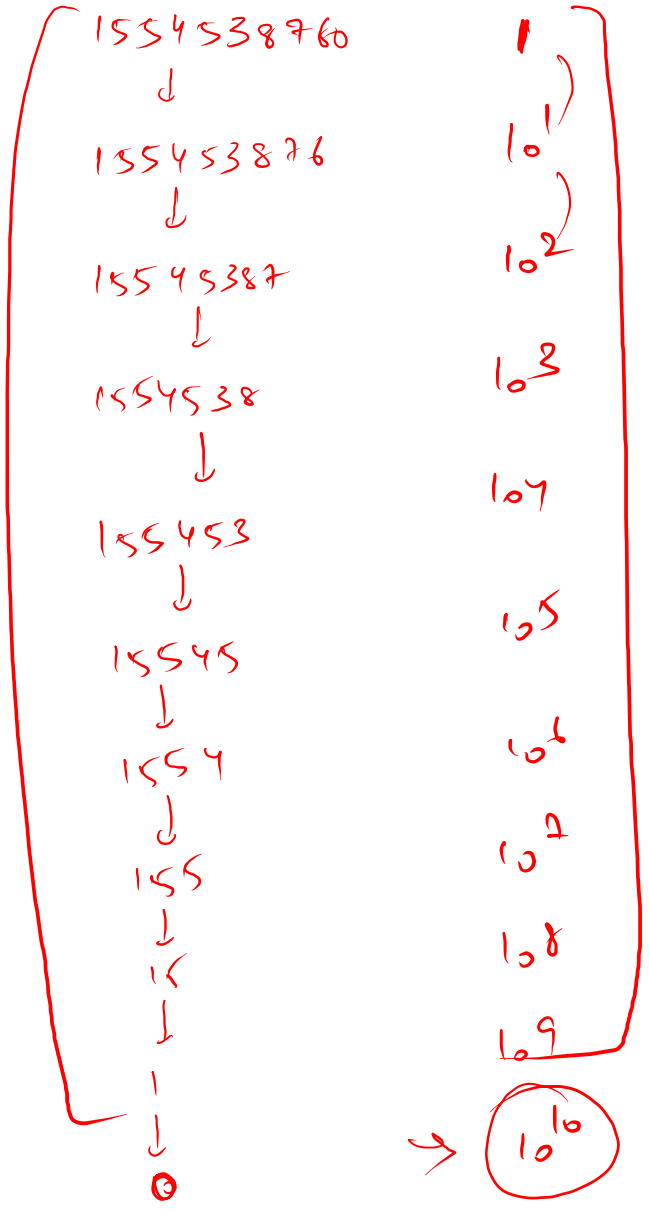

n =
1000
div
mod
1
0
tmp = 1000
100


while(n != 0){
int digit = n / tmp;
System.out.println(digit);
n = n % tmp;
tmp = tmp / 10;
}


n
1000
0
0
0
0
tmp
1000
100
10
1
0
digit
1
0
0
0


while(tmp != 0){
int digit = n / tmp;
System.out.println(digit);
n = n % tmp;
tmp = tmp / 10;
}

1554538760




```
int tmp = 1;
while(copyOfN != 0){
  copyOfN = copyOfN / 10;
  tmp = tmp * 10;
}
tmp = tmp / 10;
```

2147483647

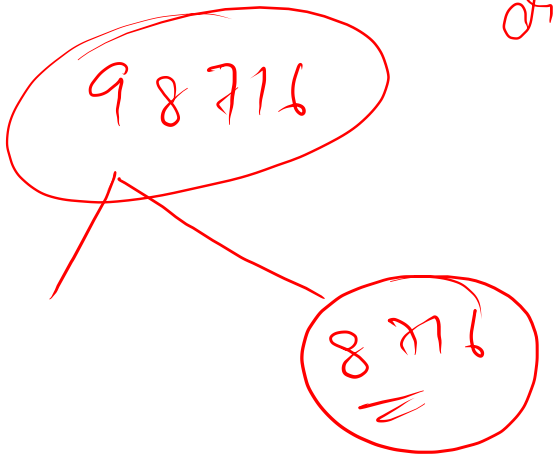
1000000000

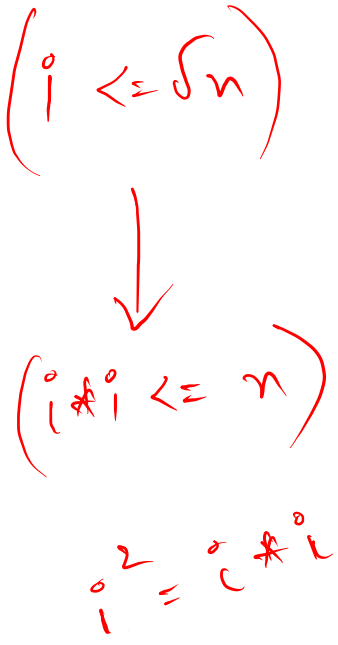
tmp ⇒ 10^9

overflow → wrong ans

predict X

```
while(div>0)// n>0 will not w
{
    int fst=n/div;
    System.out.println(fst);
    div=div/10;
    n=n%div;
}
```




i <= sqrt n
i * i <= n
i^2 = i * i