Print all digits from end

o/p

8
7
0
1
5
3
2

n = 2 3 5 1 0 7 8

```
while ( n > 0 ) {
    int rem = n % 10;
    Sys.o (rem);
    n = n / 10;
}
```

dry run

n > 0 , rem = 8

n = 2 3 5 1 0 7 > 0 , rem = 7

n = 2 3 5 1 0 > 0 , rem = 0

n = 2 3 5 1 > 0 , rem = 1

n = 2 3 5 > 0 , rem = 5

n = 2 3 > 0 , rem = 3

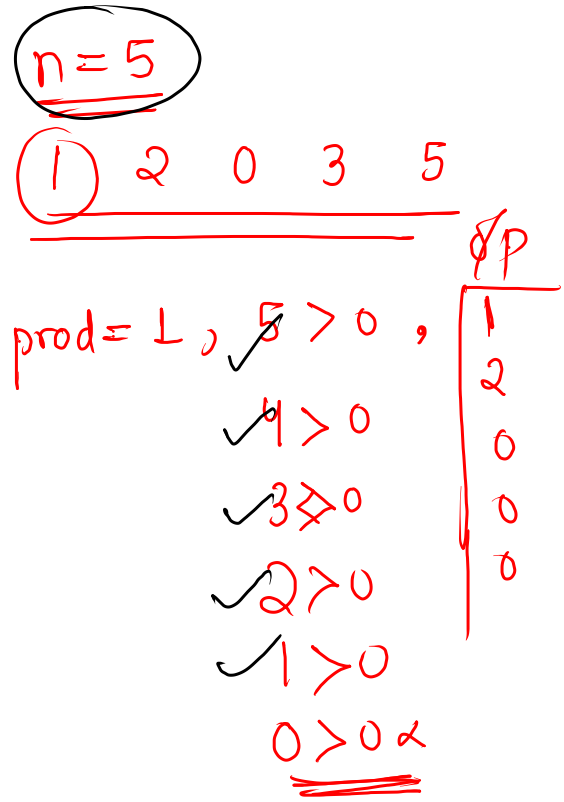
n = 2 > 0 , rem = 2

n = 0 > 0 false

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    while ( n > 0 ) {
        int rem = n % 10;
        System.out.println(rem);

        n /= 10;
    }
}
```

Running product while loop.

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

    int prod = 1;
    while ( n--> 0 ) {
        int num = scn.nextInt();
        prod = prod * num;

        System.out.print(prod + " ");
    }
}
```



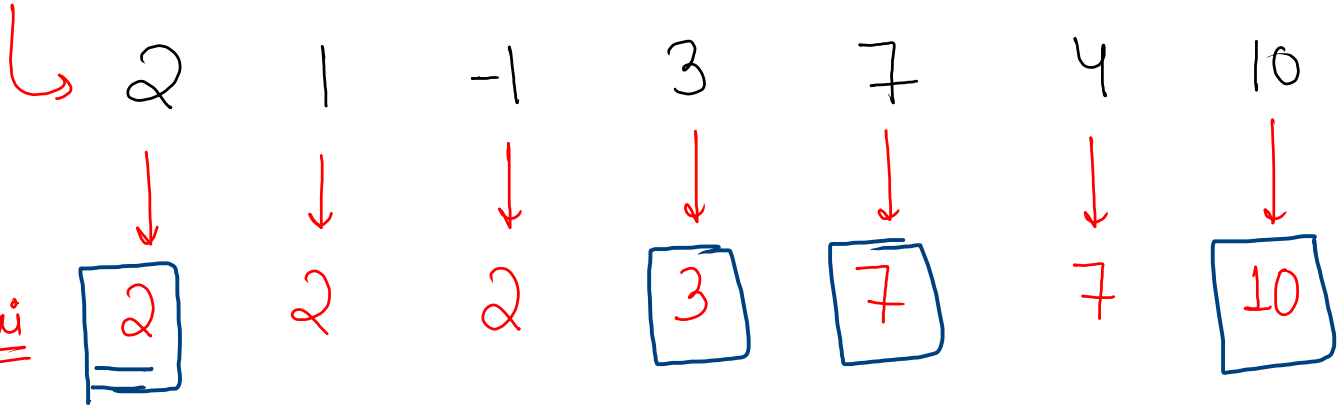
Print total steps when n/2

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

    int steps = 0;
    while ( n >= 1 ) {
        steps++;
        n = n / 2;
    }
    System.out.println(steps);
}
```

Print steps and update maximum

$n = 7$

maximal means max till now



maxi = $-\infty$

Steps = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

Print steps and update maximum

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

    int steps = 0;
    int maxi = -100;
    while ( n-- > 0 ) {

        int num = scn.nextInt();
        if ( num > maxi ) {
            steps++;
            maxi = num;
        }

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

n = 7

steps = 0
maxi = -100

num = 2 , (2 > -100) , step=1 , maxi = 2
num = 1 , (1 > 2) , step=1 , maxi = 2
num = -1 , (-1 > 2) , step=1 , maxi = 2
num = 3 , (3 > 2) , step=2 , maxi = 3
num = 7 , (7 > 3) , steps = 3, maxi = 7
num = 4 , (4 > 7) , steps = 3, maxi = 7
num = 10 , (10 > 7) , steps = 4, maxi = 10

Print nth Tribonacci number

n = 7

0	1	2	3	4	5	6	7
0	1	1	2	4	7	13	24 .....
				a	b	c	sum

logic

```
sum = a+b+c;
a = b;
b = c;
c = sum;
```

Code

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

    if ( n == 0 ) {
        System.out.println(0);
    } else if ( n == 1 ) {
        System.out.println(1);
    } else if ( n == 2 ) {
        System.out.println(1);
    } else {

        int a = 0;
        int b = 1;
        int c = 1;
        int sum = 0;

        int i = 3;
        while ( i <= n ) {
            sum = a + b + c;
            a = b;
            b = c;
            c = sum;

            i++;
        }

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

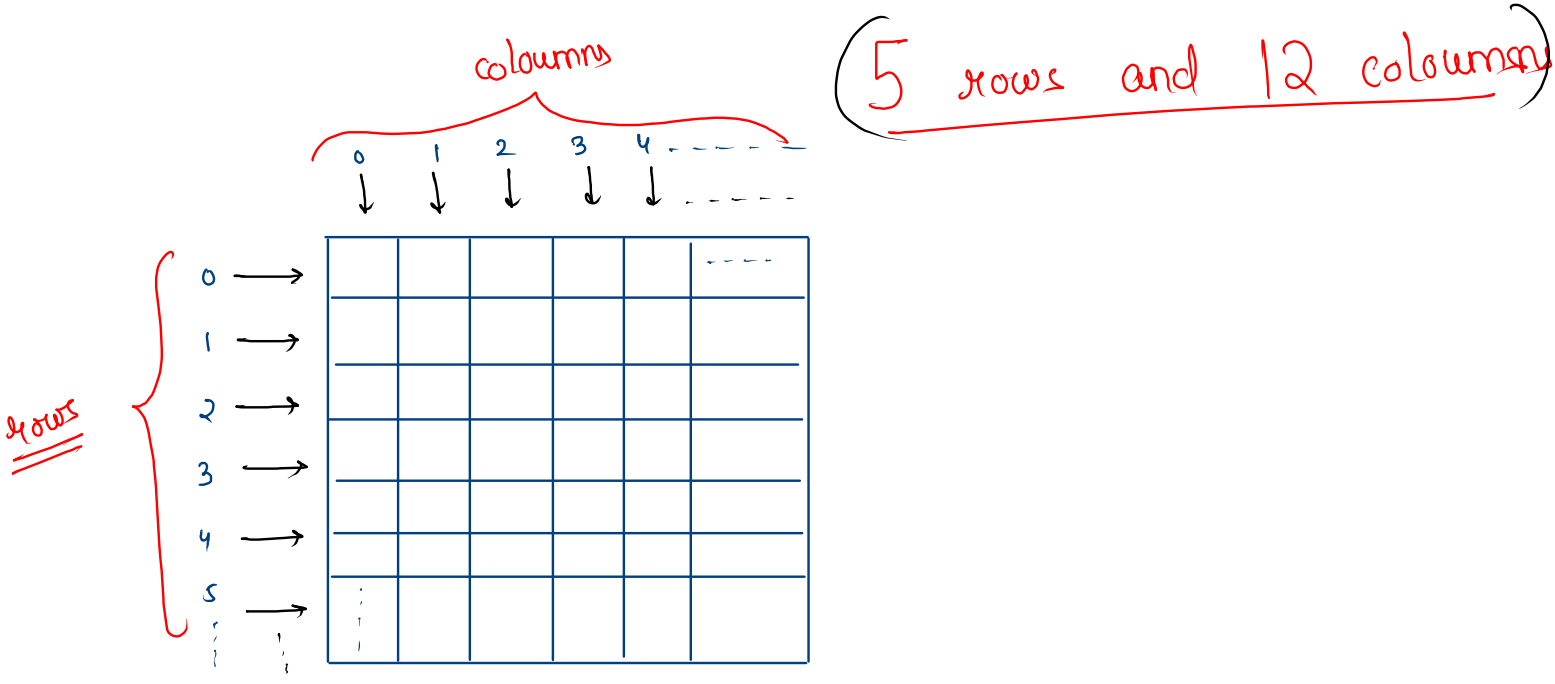
Pattern 1 - Print Stars in same line

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

    // for (int i = 0; i < n; i++) {
    //     System.out.print("*");
    // }

    int i = 0;
    while (i++ < n) {
        System.out.print("*");
    }
}
```

Nested loops

Pattern 2 - Print n x 12 star rectangle

i/p n = 5 dimention :- 5 x 12



(5 rows and 12 columns)