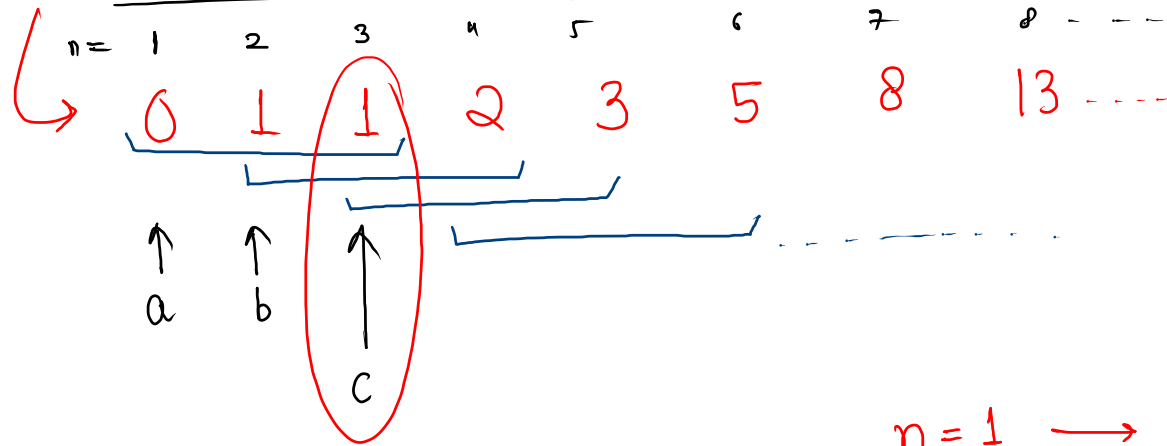


# ⇒ Fibonacci series



Imp

loop

```
c = a + b;  
a = b;  
b = c;  
syso(c);
```

n = 1 → 0  
n = 2 → 1

series :-

n=	1	2	3	4	5	6	7	
	1	1	2	3	5	8	13	-----
	↑	↑	↑					
	a	b	c					

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

```
    if ( n == 1 ) {
        System.out.println(1);
    } else if ( n == 2 ) {
        System.out.println(1);
    } else {
        → int a = 1;
        int b = 1;
        int c = 0;
        for (int i = 3; i <= n; i++) {
            c = a + b;
            → a = b;
            → b = c;
        }
        → System.out.println(c);
    }
}
```

i/p; n = 1

n = 2

n = 3, ..., ∞

n = 5

i = 3 (3 <= 5)

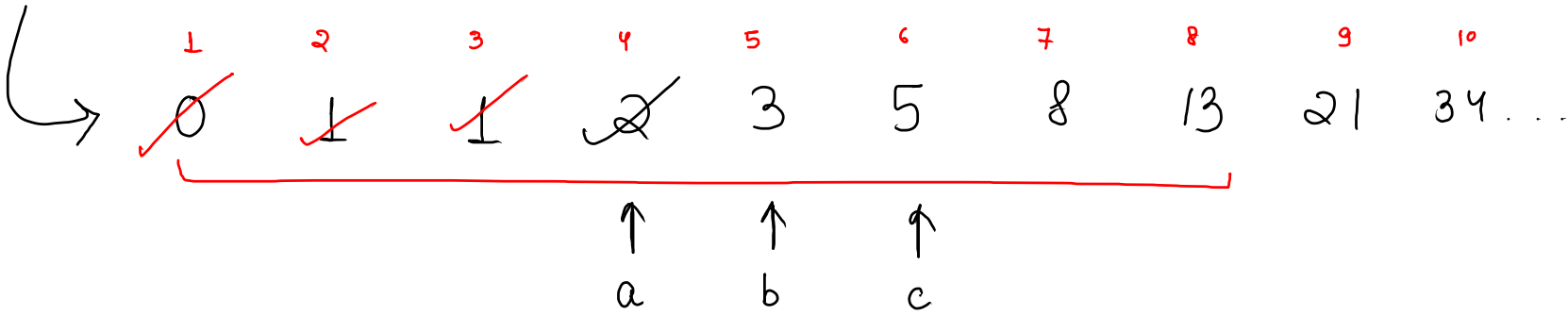
i = 4 (4 <= 5)

i = 5 (5 <= 5)

i = 6 (6 <= 5) X

	1	1	2	3	5	8	13	-----
					↑			
					c			
				↑	↑			
				a	b			

# Fibonacci number 12



i/p    n = 8

int a = 0 ;  
int b = 1 ;

o/p        0    1    1    2

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    int a = 0; ✓  
    int b = 1; ✓  
    int c = 0; ←  
    for (int i = 0; i < n; i++) {  
        System.out.print(a + " ");  
        c = a + b;  
        a = b;  
        b = c;  
    }  
}
```

#

list no.

0 1 1 2 3

i=5,

i=0, c=1

a=1

b=1

i=1, c=2

a=1

b=2

i=2, c=3

a=2

b=3

i=3, c=5

a=3

b=5

i=4, c=8

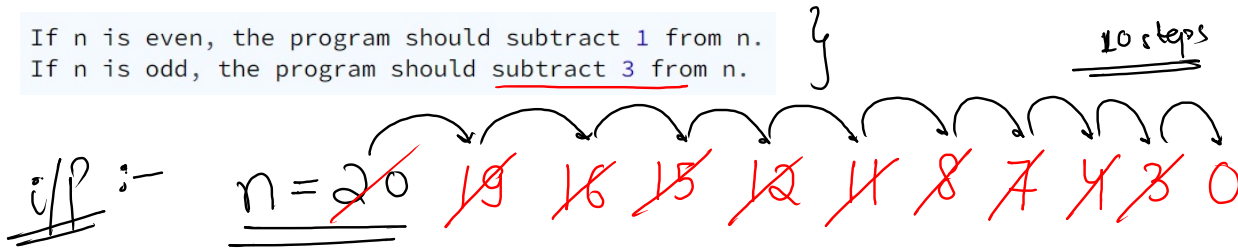
a=5

b=8

n=5

# Steps till n greater than 0

If n is even, the program should subtract 1 from n.  
If n is odd, the program should subtract 3 from n.



```
while ( n > 0 ) {  
    if ( n % 2 == 0 ) {  
        even  
        n--;  
    } else {  
        odd  
        n = n - 3;  
    }  
    count++;  
}
```

sys0 (count)

t = 3

n = 5 , → 3 ←  
n = 20 , → 2 ←  
n = 37 , → 1 ←

# Steps till n greater than 0

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int t = scn.nextInt();  
  
    for (int i = 0; i < t; i++) {  
        int n = scn.nextInt();  
        int steps = 0;  
        while ( n > 0 ) {  
            if ( n % 2 == 0 ) {  
                n--;  
            } else {  
                n -= 3;  
            }  
            steps++;  
        }  
        System.out.println(steps);  
    }  
}
```

# nth power of 10 using while loop

$$10^n = 10 * 10 * 10 * \dots$$

n times

$$\text{int ans} = 1 * 10 * 10 * \dots$$

n times

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    int ans = 1;  
    int i = 0;  
    while ( i < n ) {  
        ans = ans * 10;  
        i++;  
    }  
    System.out.println(ans);  
}
```

3 Steps

↳ understand que  
↳ logic/ dry run  
↳ code



# GKSTR46 Number of Digits

↳  $n = \underline{2315462}$

$c = 0$

step1      $n/10$  ,     231546      $c = 1$

step1      $n/10$  ,     23154      $c = 2$

step1      $n/10$  ,     2315      $c = 3$

step1      $n/10$  ,     231      $c = 4$

step1      $n/10$  ,     23      $c = 5$

step1      $n/10$  ,     2      $c = 6$

step1      $n/10$  ,     0      $c = 7$

int count = 0

while( $n > 0$ ) {

$n = n/10;$

count++;

}

code

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