Code

```
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 sum = 0;
        for (int i = 3; i <= n; i++) {
            sum = a + b;
            a = b;
            b = sum;
        }
        System.out.println(sum);
    }
}
```

a

b

c

2

False

int n = 4

a = 1

b = 1

sum = 0

i = 3, (3 <= 4) ✓

sum = 2

a = 1

b = 2

i = 4, (4 <= 4) ✓

sum = 3

a = 2

b = 3

i = 5, (5 <= 4) X

Fibonacci number 12

n = 10

	1	2	3	4	5	6	7	8	9	10
series :-	0	1	1	2	3	5	8	13	21	34
	↑ a	↑ b	↑ sum							

loop running	term
1	1
2	2
3	3
⋮	⋮
n^{th}	n^{th}

Code

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

    int a = 0;
    int b = 1;
    int sum = 0;
    for (int i = 0; i < n; i++) {
        System.out.print(a + " ");
        sum = a + b;
        a = b;
        b = sum;
    }
}
```

o/p

0 1 1 2 3 5 8 13 21 34

int n = 10

a = 0 1 1 2 3 5 8 13 21 34 55

b = 1 1 2 3 5 8 13 21 34 55 89

sum = 0 1 2 3 5 8 13 21 34 55 89

i = 0, (0 < 10) ✓

i = 1, (1 < 10) ✓

i = 2, (2 < 10) ✓

i = 3, (3 < 10) ✓

i = 4, (4 < 10) ✓

i = 5, (5 < 10) ✓

i = 6, (6 < 10) ✓

i = 7, (7 < 10) ✓

i = 8, (8 < 10) ✓

i = 9, (9 < 10) ✓

i = 10, (10 < 10) ✗

Steps till n greater than 0

$t = 2$

- $n = 20$ ✓
- $n = 37$ ✓

$n = 20$; from n to 0 by -1 if n is even and by -3 if n is odd

Code

```
int n = scn.nextInt();
int steps = 0;
while(n>=0){
    if(n%2==0){
        n--;
    }else{
        n-=3;
    }
    steps++;
}
```

Code

```java
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);
    }
}
```

t = 2

n = 20 → 10

n = 37 → 19

dry run

n = 20

steps = ~~0 1 2 3 4 5 6 7 8 9~~ 10

n > 0

20 > 0 , even
19 > 0, odd
16 > 0, even
15 > 0, odd
12 > 0, even
11 > 0, odd
8 > 0, even
7 > 0, odd
4 > 0, even
3 > 0, odd
0 > 0, ×

nth power of 10 using while loop

int n = 3

$ans = 10^3 = 10 * 10 * 10$

psudo code

from 0 to <n by +1

```
int n = sc.nextInt();
int ans = 1;
int i = 0;
while (i<n){
    ans = ans * 10;

    i++;
}
```

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

    int i = 0;
    int ans = 1;
    while (i < n) {
        ans = ans * 10;
        i++;
    }

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

int n = 3;

ans = 1

i = 0

i = 0 < 3 , ans = 10

i = 1 < 3 , ans = 100

i = 2 < 3 , ans = 1000

i = 3 < 3 ✗

$10^3 = 1000$

Running product while loop.

↳ product of all previous no. including itself

n = 5

	running prod.
↳ num = 2	2
↳ num = 3	6
↳ num = 1	6
↳ num = 0	0
↳ num = 100	0

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int ans = 1;

    for (int i = 0; i < n; i++) {
        int num = scn.nextInt();
        ans = ans * num;
        System.out.print(ans + " ");
    }
}
```