# Print nth Tribonacci number

↳ every element is sum of previous 3 no.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|---|
| series :- | 0 | 1 | 1 | 2 | 4 | 7 | 13 | ..... |
| | ↑ | ↑ | ↑ | ↑ | | | | |
| | a | b | c | sum | | | | |

| term ⟶ | loop run |
|---|---|
| 3 | 1 |
| 4 | 2 |
| 5 | 3 |
| ⋮ | |
| $n^{th}$ | $(n-2)^{th}$ |

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

(n-2) times

# Print all digits from end

n = 153242

int rem = n % 10 , n = n / 10 ;

int rem = 153242 % 10 = 2 , n = 153242 / 10 = 15324 > 0

int rem = 15324 % 10 = 4 , n = 15324 / 10 = 1532 > 0

int rem = 1532 % 10 = 2 , n = 1532 / 10 = 153 > 0

int rem = 153 % 10 = 3 , n = 153 / 10 = 15 > 0

int rem = 15 % 10 = 5 , n = 15 / 10 = 1 > 0

int rem = 1 % 10 = 1 , n = 1 / 10 = 0 X

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

# GKSTR46 Number of Digits

n = 12345

ans = 5

digits = 0

n > 0 , n = 12345 / 10 , digits = 1

n > 0 , n = 1234 / 10 , digits = 2

n > 0 , n = 123 / 10 , digits = 3

n > 0 , n = 12 / 10 , digits = 4

n > 0 , n = 1 / 10 , digits = 5

n > 0 , X

code

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

    int digit = 0;
    while ( n > 0 ) {
        n = n / 10;
        digit++;
    }

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

# Print total steps when n/2

n = 33 ,

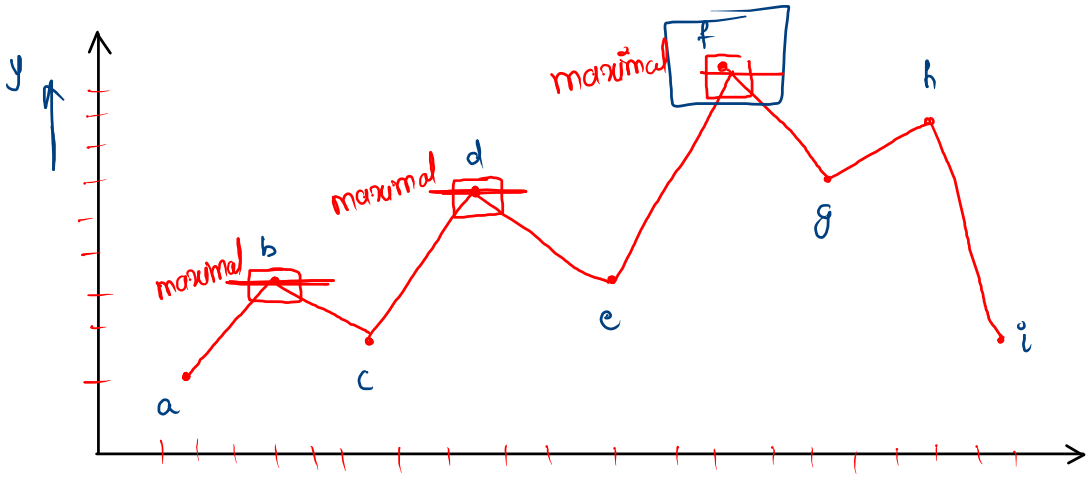
| n >= 1 , | n = n/2 , | steps = 0 |
|---|---|---|
| 33 >= 1 , | n = 16 , | steps = 1 |
| 16 >= 1 , | n = 8 , | steps = 2 |
| 8 >= 1 , | n = 4 , | steps = 3 |
| 4 >= 1 , | n = 2 , | steps = 4 |
| 2 >= 1 , | n = 1 , | steps = 5 |
| 1 >= 1 , | n = 0 , | steps = 6 |
| 0 >= 1 X | | |

Code

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

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

# Print steps and update maximum




maximum :- greatest value

maximal :- greatest value till now

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

    int count = 0;
    int maximal = -100;
    for (int i = 0; i < n; i++) {
        int num = scn.nextInt();

        if ( num > maximal ) {
            count++;
            maximal = num;
        }
    }
    System.out.println(count);
}
```

n = 6

count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

maxi = ~~-100~~ ~~1~~ ~~3~~ ~~7~~ 9

num = 1 , (1 > -100) ✓

num = 3 , (3 > 1) ✓

num = 2 , (2 > 3) ✗

num = 7 , (7 > 3) ✓

num = 5 , (5 > 7) ✗

num = 9 , (9 > 7) ✓