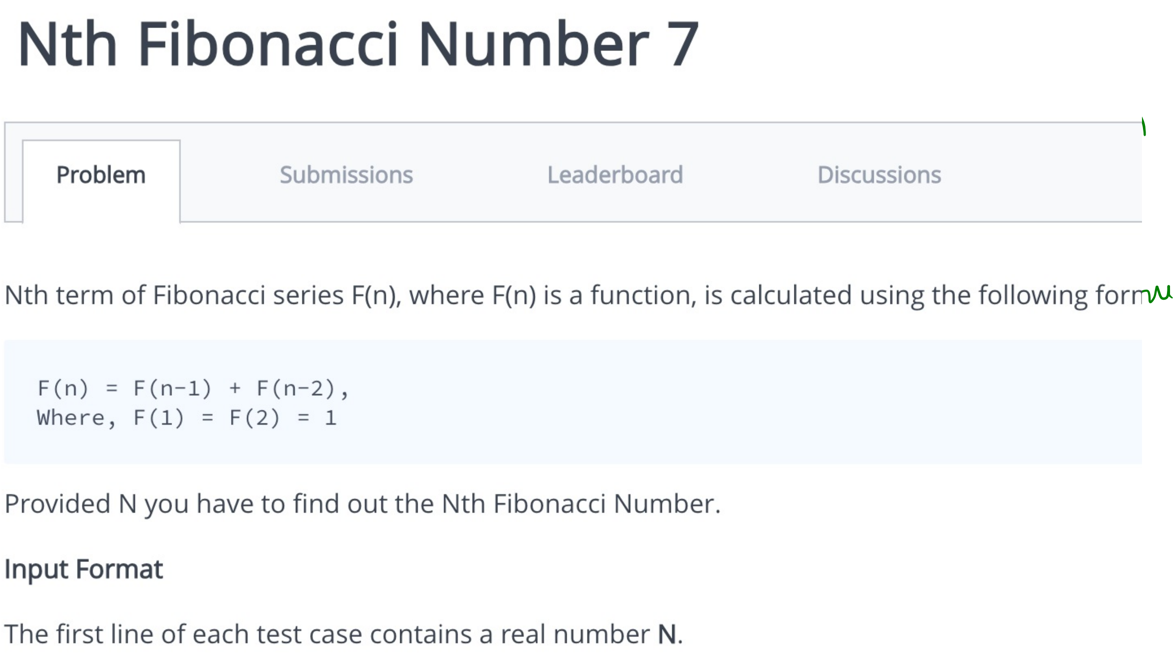


Nth Fibonacci Number 7

Problem Submissions Leaderboard Discussions

Nth term of Fibonacci series F(n), where F(n) is a function, is calculated using the following formue

//logic.

```
F(n) = F(n-1) + F(n-2),
Where, F(1) = F(2) = 1
```

Provided N you have to find out the Nth Fibonacci Number.

Input Format

The first line of each test case contains a real number **N**.

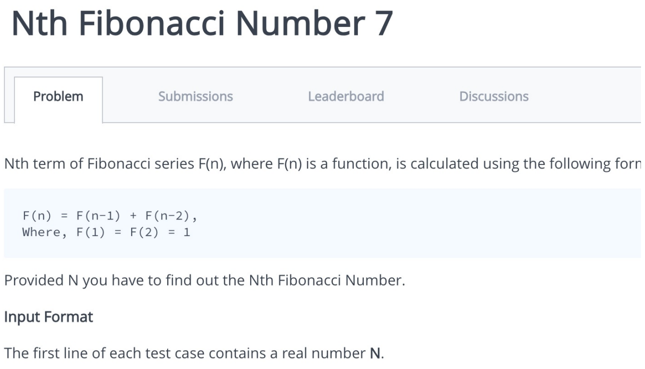
false.

base number.

↳ 0 1 1 2 3 5 8 13 . . . .

↳ 0 1 1 2 3 5 8 . . . . .

$$\boxed{T_n = T_{n-1} + T_{n-2}}$$



Nth Fibonacci Number 7

Nth term of Fibonacci series F(n), where F(n) is a function, is calculated using the following forn

```
F(n) = F(n-1) + F(n-2),
Where, F(1) = F(2) = 1
```

Provided N you have to find out the Nth Fibonacci Number.

Input Format

The first line of each test case contains a real number **N**.

$F_n = F_{n-1} + F_{n-2}$

F(1) = 1

F(2) = 1

1 1

for'

n=6.
↳ 8

n=5
↳ 5

1 1 2 3 5 8 13 21 . . . . . .

~~a~~ ~~b~~ b
a

a = ~~1~~ 1 ⟶ 1st

b = ~~1~~ 2 ⟶ 2nd

```
for(
    c = a+b
    a = b
    b = c
)
```

1 1 2 3 5 8 ↓

a b c
a b c
a b c
a b c
a b

a = ~~1~~ ~~1~~ ~~2~~ ~~3~~ 5
b = ~~1~~ ~~2~~ ~~3~~ 5 8

repeat.

```
c = a+b ✓
a = b
b = c
```

a = b

b = c

b = c

fibonacci series.

↳ ↓

0 1 1 2 3 5.....

$\bar{a}$ $\bar{b}$

0 [$\bar{1}$ $\bar{1}$ 2

1 1

✓ $\boxed{T_n} = T_{n-1} + T_{n-2}$

↓

0 1 1 2 3 5 8 13 . . . .

$\boxed{-ve}$

↓

$T_n = T_{n-1} + T_{n-2}$

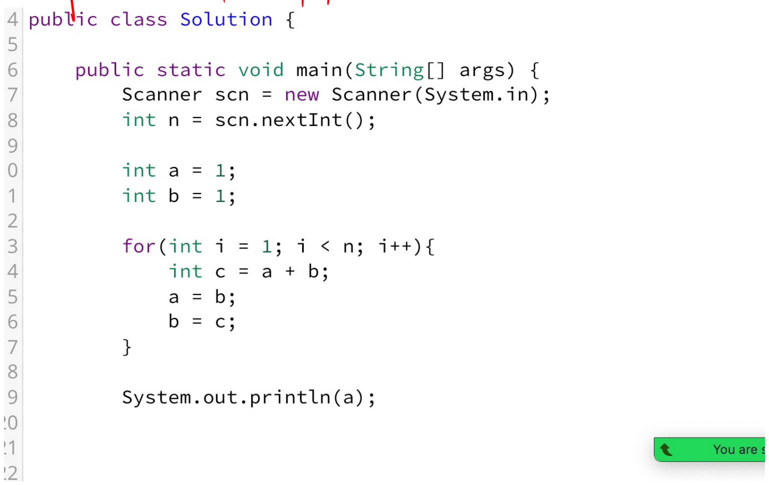
1 1

Sunday. (pen & paper)

n=6

n=7

n=5



```java
public class Solution {

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

        int a = 1;
        int b = 1;

        for(int i = 1; i < n; i++){
            int c = a + b;
            a = b;
            b = c;
        }

        System.out.println(a);
```

6th

n=6 → 8

a = ~~1~~ ~~1~~ ~~2~~ ~~3~~ ~~5~~ 8 ✓

b = ~~1~~ ~~2~~ ~~3~~ ~~5~~ ~~8~~ 13

i = ~~1~~ ~~2~~ ;
~~3~~ ~~4~~ ~~5~~ 6

a=b
b=c

1<6 ✓

2<6 ✓

c=3

3<6 ✓

c=5

4<6 ✓

c=8

5<6

c=13.

✗ 6<6

a = 8

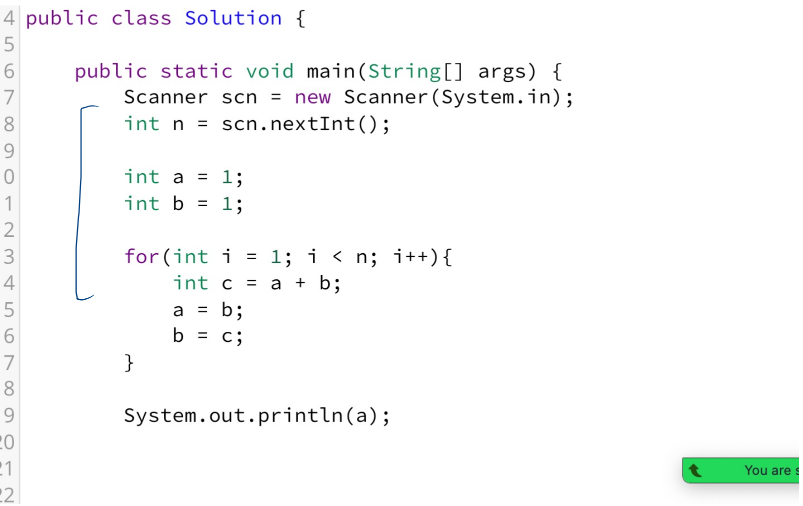
b = 13

your ref

1 1 2 3 5 8 . . . .

a @

n=6. → 8



```java
public class Solution {

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

        int a = 1;
        int b = 1;

        for(int i = 1; i < n; i++){
            int c = a + b;
            a = b;
            b = c;
        }

        System.out.println(a);
```

a=~~1~~ 1

b=~~1~~ 2

$\overset{0}{i}=1$

1<6 ✓

c=2

2

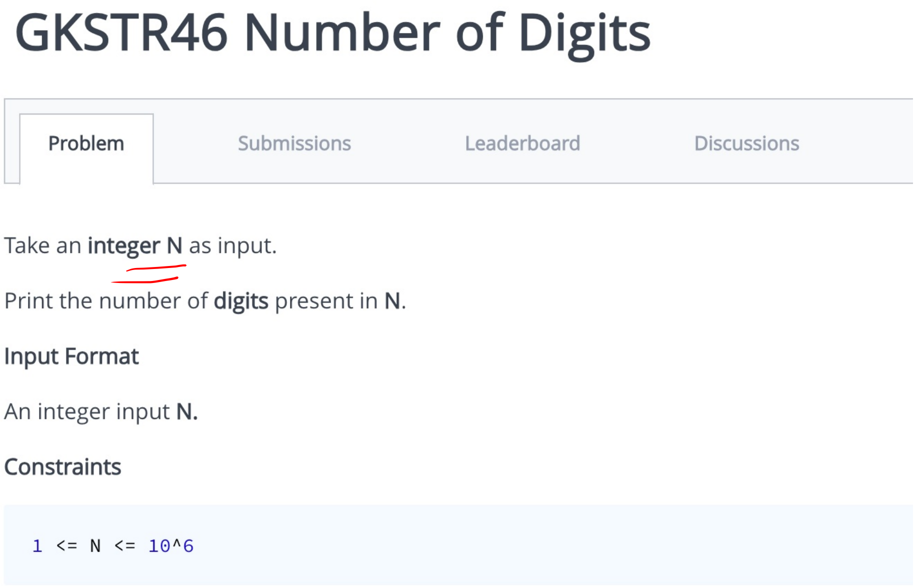
1 1 2 3

a b

a b

a = b

b = c



GKSTR46 Number of Digits

Take an **integer N** as input.

Print the number of **digits** present in **N**.

Input Format

An integer input **N.**

Constraints

```
1 <= N <= 10^6
```

N= 1234

↳ 4

print(→ 4)

N= 57609

↳ 5

?

Count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

$$\begin{tikzcd}
1234 \arrow[d, "/10"] \\
123 \arrow[d, "/10"] \\
12 \arrow[d, "/10"] \\
1 \arrow[d, "/10"] \\
\boxed{0}
\end{tikzcd}$$

eg: n = 72351

↓ /10

7235

↓ /10

723

↓ /10

72

↓ /10

7

↓ /10

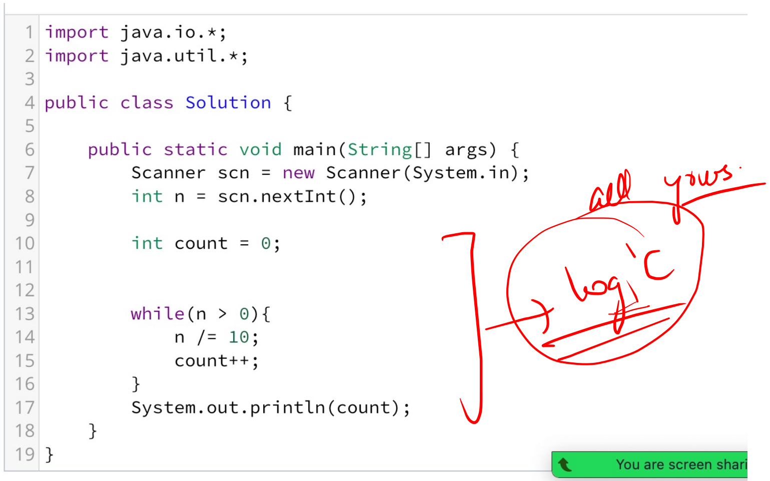
~~stop~~ 0

count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

count++

int → 0000 ✗

→ 0



```java
import java.io.*;
import java.util.*;

public class Solution {

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

        int count = 0;


        while(n > 0){
            n /= 10;
            count++;
        }
        System.out.println(count);
    }
}
```

all yours.

→ log C

n = 1234

count = ~~0~~ · ~~1~~ 2

~~1234 > 0~~

1234 / 10 = 123

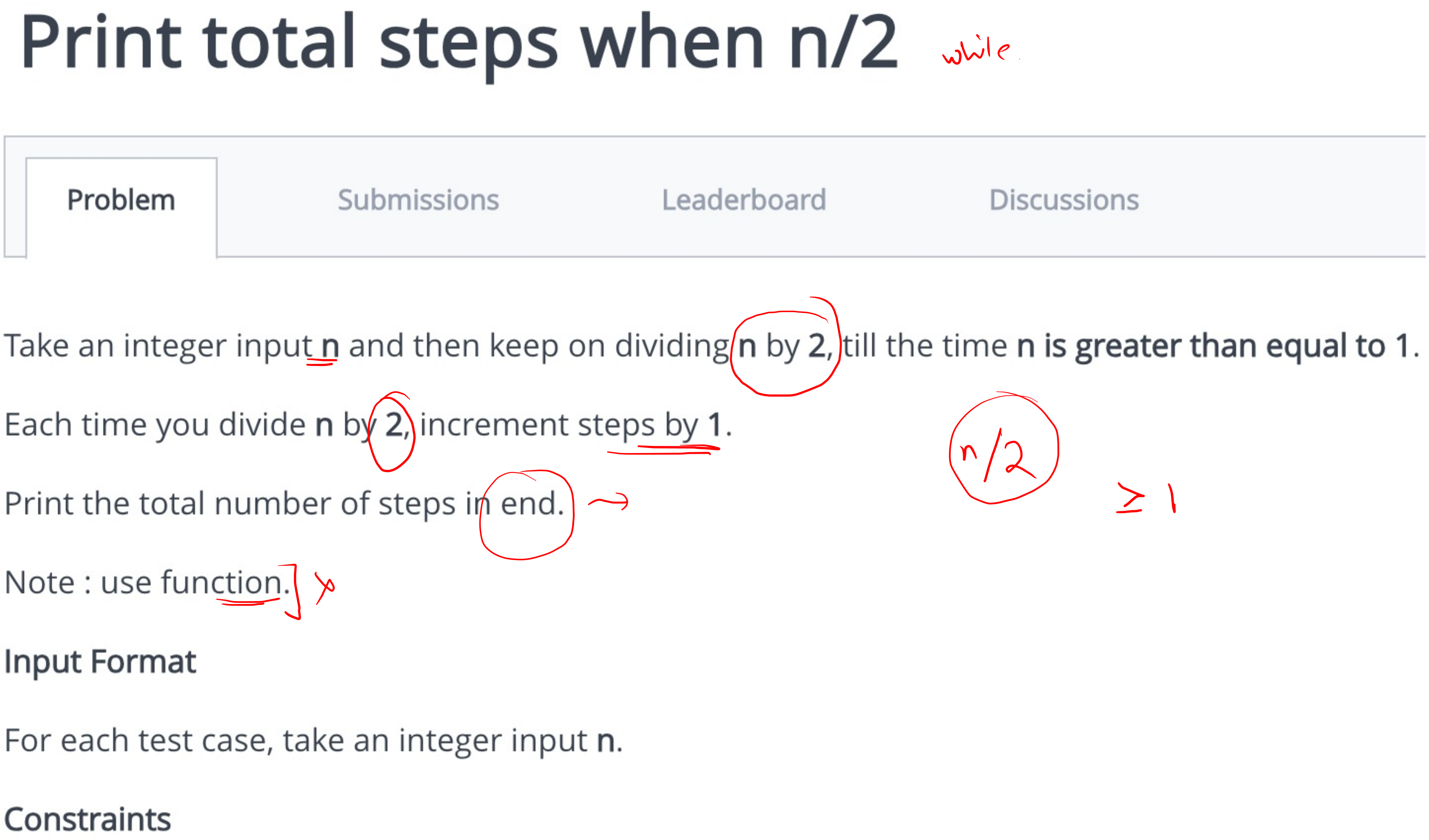
123 > 0

12 > 0

1 2 3 4

?

n → count



Print total steps when n/2

while

< 1

Take an integer input **n** and then keep on dividing **n** by **2**, till the time **n is greater than equal to 1**.

Each time you divide **n** by **2**, increment steps by **1**.

$n/2$ ≥ 1

Print the total number of steps in end. →

Note : use function.

Input Format

For each test case, take an integer input **n**.

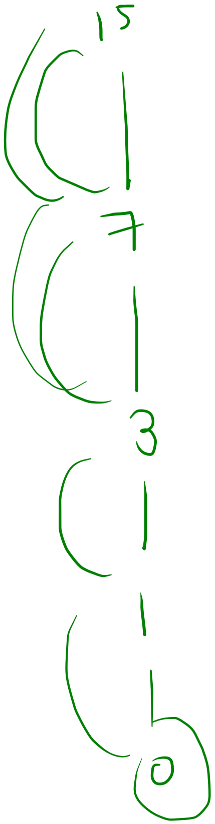
Constraints

step = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

$$\begin{tikzcd}
32 \arrow[d, "/2"] \\
16 \arrow[d, "/2"] \\
8 \arrow[d] \\
4 \arrow[d] \\
2 \arrow[d] \\
1 \arrow[r] & 0
\end{tikzcd}$$

≥ 1

< 1



≥ 1

✓

step=1 ✓

≥ 1

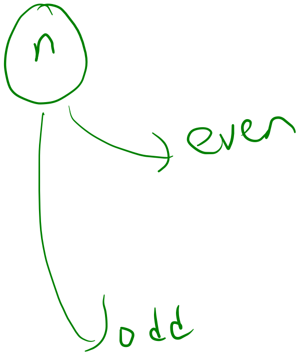
$n=0$

step = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

```
step=0
while ( $n \geq 1$ )
{
        n /= 2
        step++
}
```

$0 \geq 1$ ✗

→



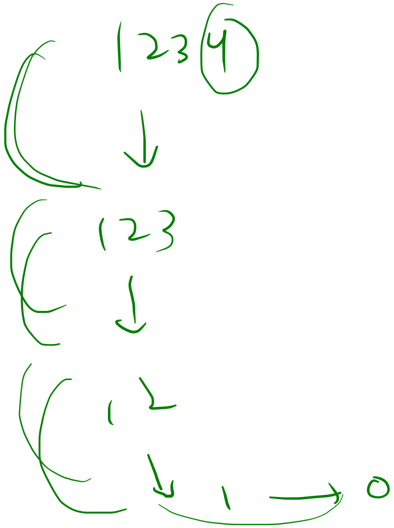
$n \% 2 == 0$ ✓

logic ✓

Maths.

steps = 4.

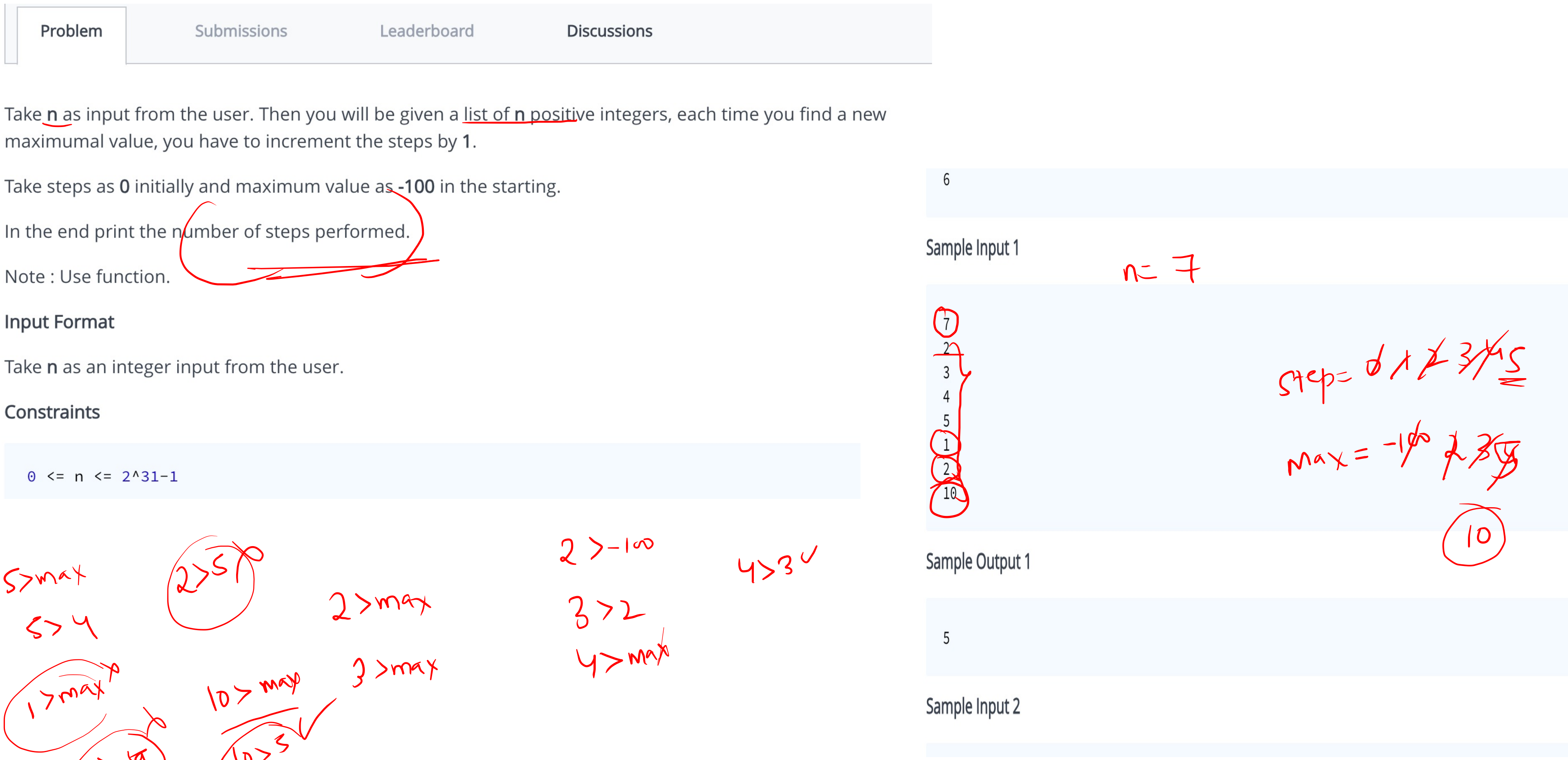
digits.



last

$n \% 10$ ✓

logic
maths.



Take n as input from the user. Then you will be given a list of n positive integers, each time you find a new maximumal value, you have to increment the steps by 1.

Take steps as 0 initially and maximum value as -100 in the starting.

In the end print the number of steps performed.

Note : Use function.

Input Format

Take n as an integer input from the user.

Constraints

```
0 <= n <= 2^31-1
```

```
6
```

Sample Input 1

n= 7

```
7
2
3
4
5
1
2
10
```

step= ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

max = ~~-100~~ ~~2~~ ~~3~~ ~~5~~

10

Sample Output 1

```
5
```

Sample Input 2

2 > -100

3 > 2

4 > max

4 > 3 ✓

5 > max

5 > 4

2 > 5 ✗

2 > max

3 > max

1 > max ✗

1 > 5 ✗

10 > max

10 > 5 ✓

5

↓ ↓ ↓
3 2 7 1 9

max = ~~-100~~ ~~3~~ ~~7~~ 9

step = ~~0~~ ~~1~~ ~~2~~ 3

i/p

compare

update.

3 > max

3 > -100

2 > max

2 > 3

7 > max

7 > 3 ✓

1 > max

1 > 7

9 > max

7

1 2 3 4 5 6 7

i/p loop
[ ]

max = -1~~∞~~ ~~3~~ ~~4~~ 7

step = ~~1~~ ~~2~~ 3

5

3 4 1 7 2

3 > max ✓

3 > -1∞

4 > max

4 > 3

1 > 4

2 > 7 ✗

7 > 4 ✓

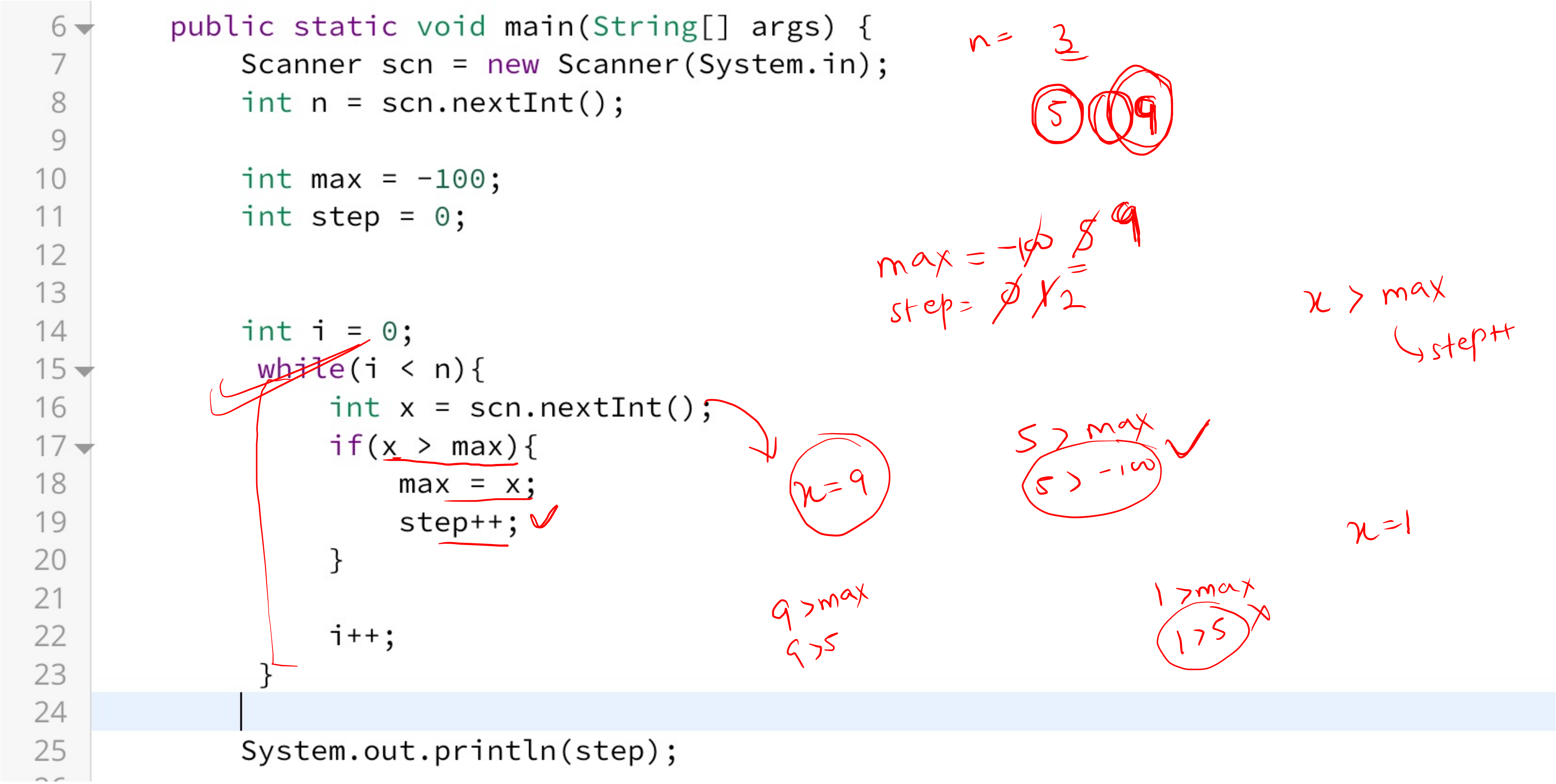
max = 4

x = 7

7 > 4

max = x

step = +1



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

        int max = -100;
        int step = 0;


        int i = 0;
         while(i < n){
            int x = scn.nextInt();
            if(x > max){
                max = x;
                step++;
            }

            i++;
         }

        System.out.println(step);
```

n = 3

5 9

max = ~~-100~~ ~~5~~ 9

step = ~~0~~ ~~1~~ 2

x > max

↳ step++

x=9

5 > max ✓

5 > -100

x=1

9 > max

9 > 5

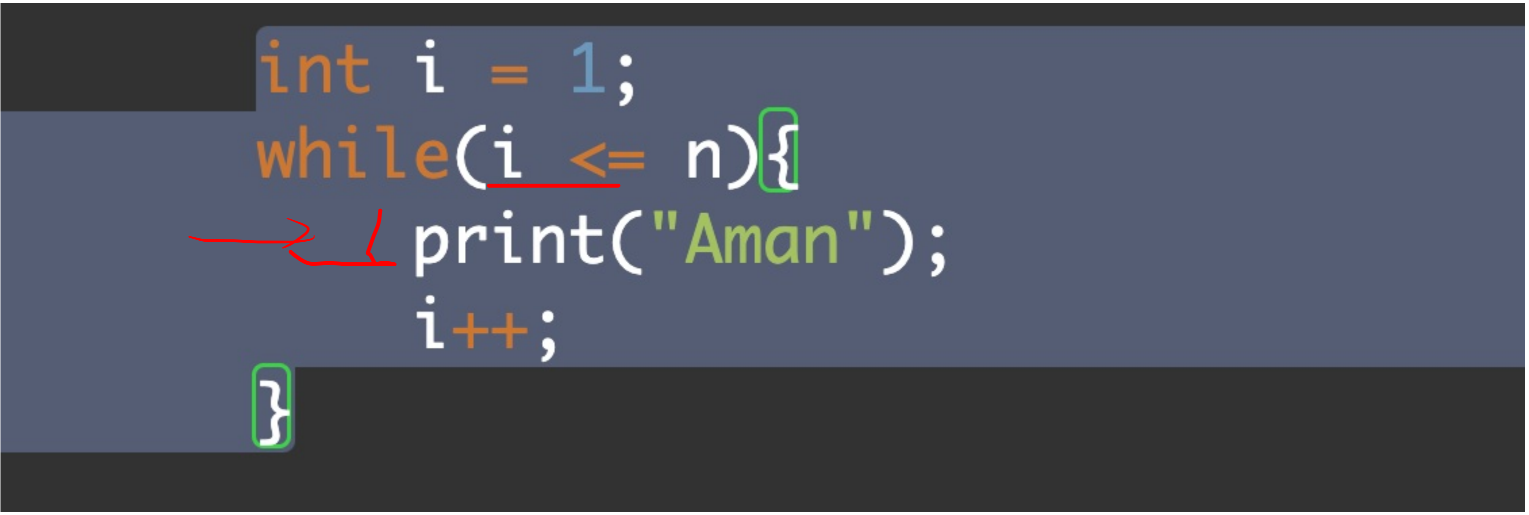
1 > max

1 > 5 ✗

No ?

No where.

n = 5



```
int i = 1;
while(i <= n){
    print("Aman");
    i++;
}
```

Aman
Aman
Aman
Aman
Aman