

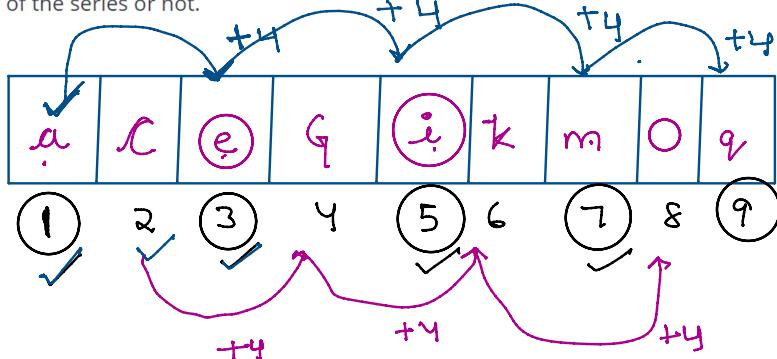
✓ MCT → 23rd Oct ✓
99%

Tomorrow → Most Important Class

Print a, C, e, G, i, K... till 'z' or 'Z'

Problem Submissions Leaderboard Discussions

Print a, C, e, G, i, K, m, O, q, S till the last character is less than 'z' or 'Z' accordingly whether 'z' or 'Z' is a part of the series or not.



char ch1 = 'a'

char ch2 = 'C'

Odd → Lowercase

Even → Uppercase

• ch1 → 'a', ch2 → 'C'

• int i = 1.

while (ch1 ≤ 'z' || ch2 ≤ 'Z') → test condn

if (i%2 != 0) → odd

print(ch1);
ch1 += 4

} lowercase

else → even

print(ch2);
ch2 += 4

uppercase

i++;

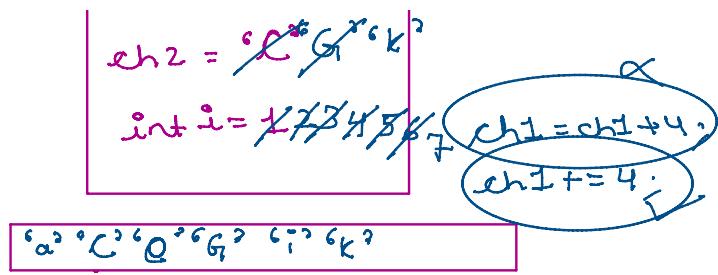
```
char ch1 = 'a';  
char ch2 = 'C';  
int i = 1;  
while(ch1 <= 'z' || ch2 <= 'Z') {  
    if(i%2 != 0){ — odd  
        System.out.println(ch1);  
    } else {  
        System.out.println(ch2);  
    }  
    i++;  
}
```

ch1 = ~~'a' 'c' 'e' 'g' 'i' 'k' 'm'~~
ch2 = ~~'C' 'G' 'K' 'M'~~

```

while(ch1 <= 'z' || ch2 <= 'Z'){
    if(i%2 != 0){ — odd
        System.out.println(ch1);
        ch1+=4;
    }else{ → even
        System.out.println(ch2);
        ch2+=4;
    }
    i++;
}

```



a, C, e, G, i, K... till 'z' or 'Z'

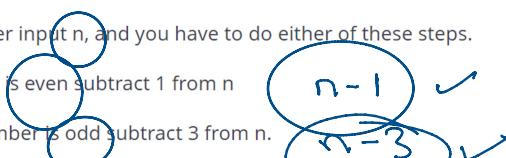
Steps till n greater than 0

Problem Submissions Leaderboard Discussions

20

Take an integer input n, and you have to do either of these steps.

If the number is even subtract 1 from n



step ++ ✓

and if the number is odd subtract 3 from n.

Keep on performing these steps till the time the value of n is greater than 0. In the end print the total number of steps performed.

while($n > 0$)

Sample Output 0

10 ✓

$2 \rightarrow 1 \rightarrow 0$
Step = 0.5

67
89
10

20 ✓ → even
19 — odd
18 → even
17 — odd
16 → even
15 — odd
14 → even
13 → odd
12 → even
11 → odd
10 → even
9 — odd
8 → even
7 → odd
6 → even
5 — odd
4 → even
3 — odd
2 → even
1 — odd
0

print
 $i = n$

→ if - else
 $i = i - 1$
step ++

else
 $i = i - 3 ;$

```

Scanner scn = new Scanner(System.in);

```

$n \rightarrow 15$ ✓
14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int steps = 0;
int i = n;
while(i>0){
    if(i%2 == 0){
        i-=1;
    }else{
        i-=3;
    }
    steps++;
}
System.out.println(steps); 7 ✓

```

$n \rightarrow 15 \checkmark$
 $\text{steps} = 9 \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6} \cancel{7}$
 $i = 15 \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6} \cancel{7}$

nth power of 10 using while loop

Problem

Submissions

Leaderboard

Discussions

$n \rightarrow \text{input}$

Take n as an integer input and print the nth of 10 as an integer output.

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int ans = (int) Math.pow(10, n); double
System.out.println(ans);
typecast

```

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); → 4
int ans = 1;
while(n > 0){
    ans *= 10;
    n--;
}
System.out.println(ans);

```

$ans = 10000$
 $n = 4 \cancel{3} \cancel{2} \cancel{1} 0$

Print nth Tribonacci number

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)

$$\begin{aligned}a &= 0 \\b &= 1\end{aligned}$$

$$c = 1$$

$$\cancel{d = a+b+c}$$

```
for (int i = 0; i < n; i++) {  
    d = a+b+c;  
    swap(a, b, c);  
}
```

nth term T_n of The Tribonacci sequence is defined as follows:

T_0 (0th term) = 0, T_1 (1st term) = 1, T_2 (2nd term) = 1, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for $n \geq 0$.

Take n as an integer inout, print the value of T_n (nth term) as an integer output.

```
Scanner scn = new Scanner(System.in);  
int n = scn.nextInt();  
int a = 0, b = 1, c = 1;  
for (int i = 0; i < n; i++) {  
    int d = a + b + c;  
    a = b;  
    b = c;  
    c = d;  
}  
System.out.println(a);
```

}
print

9 → n
10 → n
11 → n

0	1	1	2	4	7	13	24	44	81	149
---	---	---	---	---	---	----	----	----	----	-----

44

Print all digits from end

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)

Take n as an integer input from the user, then you have to print the digits of the number starting from the end to the first digit of the number where each digit should be printed in a separate line.

Sample Input 0
 $n > 0$ test condition
 $n \rightarrow \text{input}$
 7654
 $\% 10$ fetch
 point
 $n = n / 10$
 Sample Output 0

To fetch the last digit = $\%$
 To remove the last digit = $/$

4
5
6
7

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); → 7698432
while(n > 0){ int i=n
    int ans = n % 10;
    System.out.println(ans);
    n /= 10;
}
    i--
```

$n = 7698432$
 $ans = 2$

for loop
 $(int i=7; i > 0; i /= 10)$

2
3
4
8
9
6
7

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int i = n; Initialization
while(i > 0){ test condition
    int ans = i % 10;
    System.out.println(ans);
    i /= 10; update
}
```

GKSTR46 Number of Digits

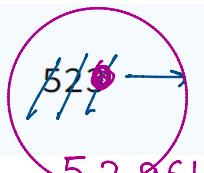
Sample Input 0

X✓3 ✓

Sample Input 0

$n \rightarrow \text{input}$

$n > 0$



Sample Output 0

3

~~digit = 0;~~ ✓
X✓3✓

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int digit = 0;
while(n > 0){
    n=n/10; → removing
    digit++; ✓
}
System.out.println(digit); — 5
```

$n \rightarrow \cancel{5}2\cancel{9}64$

$\text{digit} = \cancel{0}X\cancel{2}\cancel{3}45$

Step = ~~0~~X✓345✓

Print total steps when $n/2$

Problem Submissions Leaderboard Discussions

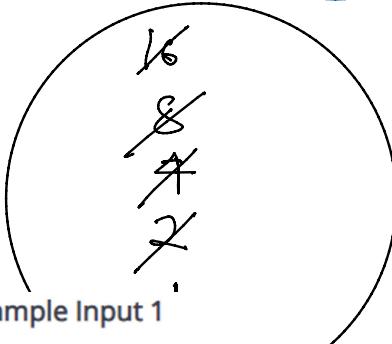
Take an integer input n on dividing n by 2, till the time n is greater than equal to 1.
Solved: 210 Attempted: 214
Each time you divide n by 2, it takes 1 step by 1.

Print the total number of steps in end.

~~32~~ → $n/1$

test condition

$n/2$



Sample Input 1

20

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int step = 0;
while(n > 0){ → int i=n
    n=n/2; → n>1
    step++; → false
}
System.out.println(step); → 0
```

step = 0
X
3
4
5

Sample Output 1

5 ✓

$20/2 \rightarrow 16\cancel{5}\cancel{2}\cancel{0}$

for (int i=n ; i>0 ; i=2)

Print steps and update maximum

Problem Submissions Leaderboard Discussions

Take n as input from the user. You will be given a list of n positive integers, each time you find a new maximum value, you print the steps by 1.

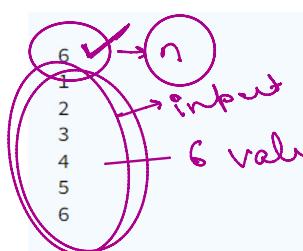
Solved: 210 Attempted: 214
Take steps as 0 initially and maximum value as -100 in the starting.

In the end print the number of steps performed.

$$\text{Step} = 0$$

$$\text{max} = -100 \checkmark$$

Sample Input 0



6 times

6 value → inside for loops
int num = scn.nextInt();
if (num > max)
max = num
step++

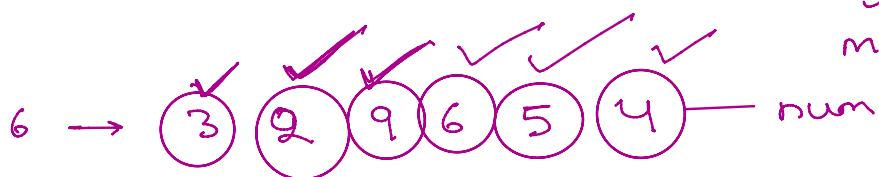
Sample Output 0

6

)
System.out.println;

$$\text{Step} = 0 \times 2 \checkmark$$

$$\text{max} = -100 \checkmark$$



3 → -100
if (num > max)
max = num

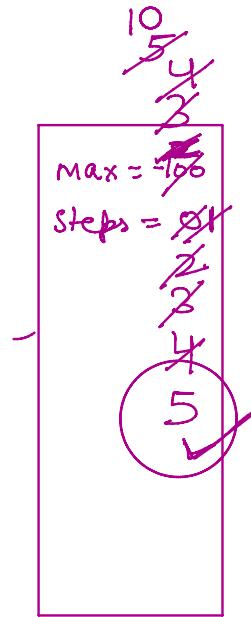
$$\text{max} = 9$$

step++

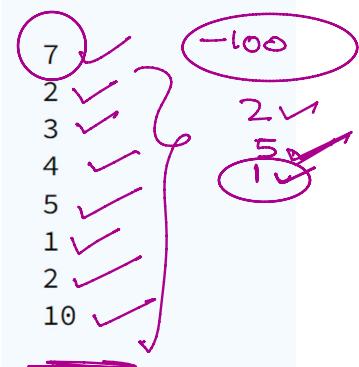
```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); → 7
int max = -100; ← 7
int steps = 0; ← i < 7
for(int i = 1; i <=n, i++){
    int num = scn.nextInt(); ← 3
    if(num > max){ ← 3 > 2
        max = num; ← 5 > 4
        steps++; ← 1 > 5
    }
}
System.out.println(steps); ← 2 > 5
                                ← 10 > 5

```



Sample Input 1



Sample Output 1

5 ✓