

## Rotate 7-digit number to right by three

Problem

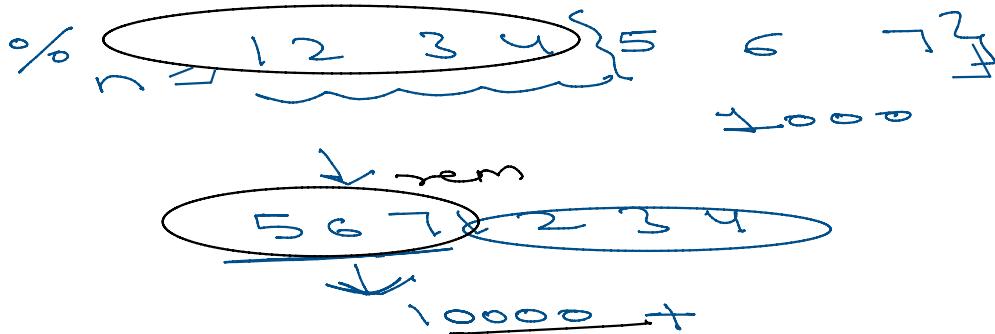
Submissions

Leaderboard

Discussions

Take n as an integer input, you have to pick the last 3 digits of the number and put them in the starting.

e.g. 1234567 is given, then this number should transform to 5671234.



```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int rem = n % 1000; // 567
n = n / 1000; // 1234
int ans = rem * 10000 + n; // 5670000+1234 = 5671234
System.out.println(ans);
```

### \* Prime Number

- $23 \rightarrow 1, 23$        $16 \rightarrow 1, 2, 4, 8, 16$
  - $5 \rightarrow 1, 5$
  - $36 \rightarrow 1, 2, 3, 6, 9, 12, 18, 36$
- $i = 2$       some  $i$   
 $\text{for } (i = 2; i < n/2; i++)$   
 $(n \% i) == 0 \rightarrow \text{return false}$
- $\Rightarrow \text{return false}$

```
static boolean isPrime(int n){
    for(int i = 2; i < n / 2; i++){
        if(n % i == 0){  $n = 37$ 
            return false;
        }
    }
    return true;
}
```

$i = 2$   
 $n = 37$   
 $i > 18$

```
public static void main(String[] args) {  
    /* Enter your code here. Read input from  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    if(isPrime(n))  
        System.out.println("Yes");  
    else  
        System.out.println("No");  
}
```

12

## Print all factors of a number

Problem	Submissions	Leaderboard	Discussions
Take a whole number n as an integer input and print all the factors of it such that each factor should be printed in a separate line.			

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); 16
for(int i =1; i<16; i++){
    if(n % i ==0){
        System.out.println(i);
    }
}
```

## Sample Output 0

12

7 2 5 8 16

## Print "prime" or "not prime"

Take n as an integer input from the user, after this n integer inputs will be given by the user. And for each integer input, print in a separate line and you have to print "prime" if the integer is a prime number and "not prime" if the integer is not a prime number.

Eg -

$\sqrt{36}$

$36 = 1 \times 36$

$18 = 2 \times 9$

$12 = 3 \times 4$

$9 = 3 \times 3$

$6 = 2 \times 3$

factors of that number

not

$$\sqrt{49} \rightarrow 7$$

~~$\pm 2, 7$~~        $2 - \sqrt{5}$

\* Optimal way

~~$\sqrt{63} = 7 \pm 1$~~

\* Time limit exceeded

~~$3 \sqrt{ }$~~

$$\sqrt{49} = 7$$

$$7 \times 5 = 35$$

$$5 \times 3 = 15$$

$$\sqrt{99} = 9$$

for (int i = 2; ~~i <= n~~; i++)

$$i \leq \sqrt{n}$$

$$i \leq (n)^{1/2}$$

$$i^2 \leq n \quad \text{and} \quad i^2 \leq n \leq \sqrt{n}$$

$$36 \rightarrow 6 \quad 49 - 7$$

$$(\sqrt{43} = 6)$$

for (i = 2, i <= 6)

$$(\text{if } i == 0)$$

prime

□ → on 10th

$$36 \rightarrow 2 \quad \text{not prime}$$

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();

for(int i = 1 ; i<=n ; i++){
    int num = scn.nextInt();
    boolean flag = false;
    for(int k = 2 ; k * k <= num ; k++){
        if(num % k == 0){
            System.out.println("not prime");
            flag = true;
            break;
        }
    }
    if(flag == false){
        System.out.println("prime");
    }
}

```

Sample Input 0

3  
45  
17  
32

*n num =  
flag = false*

Sample Output 0

not prime  
prime  
not prime

}

## Divide n by 2 3 5 and tell steps

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Take a natural number n as an integer input, and variable steps of integer type as input. Then perform the following operations on it.

- a. If the number is divisible by 2, then keep on dividing the number n by 2, till the time the number is divisible by 2 and also increment the variable steps by 2, each time you divide the number by 2.
- b. Also, check if the number is divisible by 3, then keep on dividing the number n by 3, till the time the number is divisible by 3 and also increment the variable steps by 3, each time you divide the number by 3.
- c. Also, If the number is divisible by 5, then keep on dividing the number n by 5, till the time the number is divisible by 5 and also increment the variable steps by 5, each time you divide the number by 5.

In the end print the value of the variable steps in the first line and final value of number n in the second line.

$$\begin{aligned}
 n &= \cancel{700} \quad \cancel{350} \quad \cancel{175} \quad \cancel{35} \quad \cancel{7} \\
 \text{steps} &= \cancel{0} \quad \cancel{2} \quad \cancel{4} \quad \cancel{9} \quad \cancel{14} \quad \checkmark
 \end{aligned}$$

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int steps = scn.nextInt();
if(n%2 ==0){
    while(n%2==0){
        n=n/2;
        steps +=2;
    }
}
if(n%3 ==0){
    while(n%3==0){
        n=n/3;
        steps +=3;
    }
}
if(n%5 ==0){
    while(n%5==0){
        n=n/5;
    }
}

```

Sample Input 4

700  
0

Sample Output 4

14  
7

$$\begin{aligned}
 n &\rightarrow \cancel{350} \quad \cancel{175} \quad \cancel{35} \quad \cancel{7} \\
 \text{steps} &= 6 + 2 = 8 + 5 = \\
 &\quad 13 + 5 \\
 &= 18 \quad \checkmark
 \end{aligned}$$

```

if(n%5 ==0){
    while(n%5==0){
        n=n/5;
        steps +=5;
    }
}
System.out.println(steps);
System.out.println(n);

```

## Print all unique prime factors

Problem

Submissions

Leaderboard

Discussions

Take a whole number  $n$  as an integer input and then print all the unique prime factors of  $n$  such that each prime factor is printed in a separate line.

~~45~~  $\circlearrowleft 3^2, 5^1$

~~90~~  $\circlearrowleft 3^2 \times 2^1 \times 5^1$

~~120~~  $\circlearrowleft 2^3, 5^1, 3^1$

~~3~~,  $\circlearrowleft 5^1, 11^1$  ✓

~~55~~

~~11~~ ✓

~~320~~  $\circlearrowleft 2^5, 5^1, 11^1$

~~400~~  $\circlearrowleft 2^4$  ✓

~~55~~  $\circlearrowleft 5^1$  ✓

for ( $i=2$ ) num ; )

if (num % i == 0)

print (i); ✓

while (num % i == 0)

num = num / i;

~~6~~  $\circlearrowleft$  prime

~~2~~  $\circlearrowleft$  even

~~2~~  $\circlearrowleft 48$  ✓

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); → 96
for(int i = 2; i <= n; i++) {
    if(n % i == 0){ → 96 / 2 → Yes
        System.out.println(i); →
        while(n % i == 0){ → 48 % 2 == 0
            n = n / i; → 48
            24 % 2 == 0
            12 % 2 == 0
            6 % 2 == 0
            3 % 2 == 0 →
            false
        }
    }
}

```

2, 3