

Break Statement.

```
main.java
1 import java.util.*;
2
3 public class Main
4 {
5     public static void main(String[] args) {
6
7         for(int i = 0; i < 5; i++) {
8             if(i == 1){
9                 break;
10            }
11            System.out.println(i);
12
13        }
14        System.out.println("Hello World");
15    }
16 }
17
18
```

0
1
2

$i = 0$

1

2

$0 \leq 5$ ✓

$1 \leq 5$ ✓

$2 \leq 5$ ✓

$i = 2$

$2 == 2$

```
0
1
Hello World
2
```

HCF / GCD

?

Highest
Common
factor

Greatest
Common
Divisor

Find GCD 3

Problem

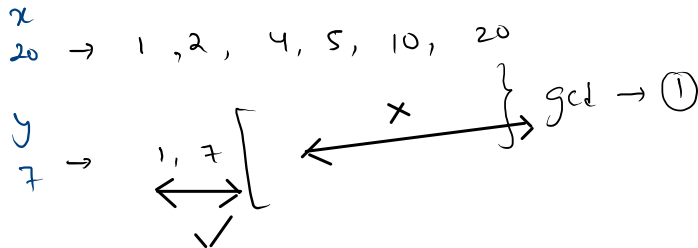
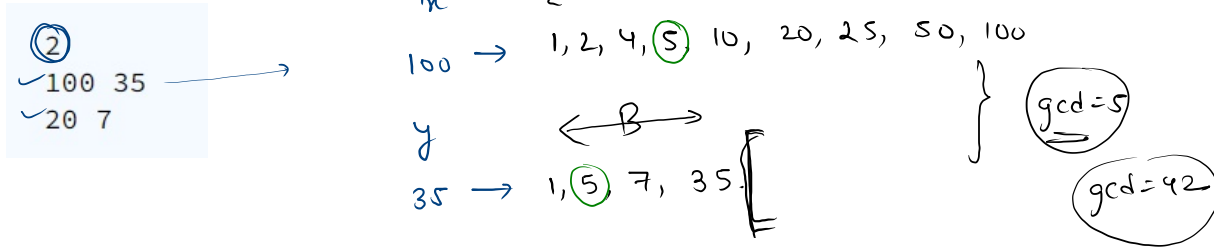
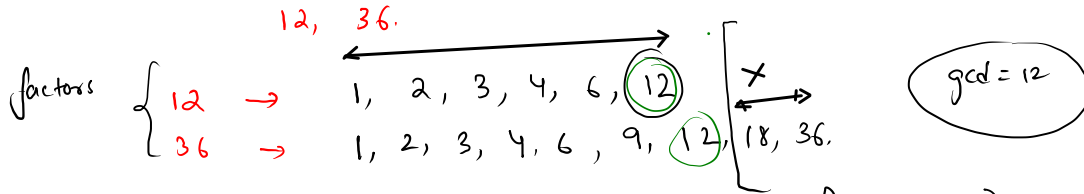
Submissions

Leaderboard

Discussions

what is the greater
factor which can divide
both?

Take two integer inputs x and y . Then print the **gcd** of these two numbers as an integer output.



x. 12 \rightarrow 1, 2, 3, 4, 6, 12

y. 36 \rightarrow 1, 2, 3, 4, 6, 9, 12, 18, 36

as = ~~1~~ ~~2~~ ~~3~~
~~4~~ ~~6~~
12

for (1 $\xrightarrow{\leq}$ min(x, y))

{ if (x % i == 0 & y % i == 0)

{ i

}

12 \rightarrow 1, 2, 3, 4, 6, 12
9 \rightarrow 1, 3, 9

```
int x = scn.nextInt();
int y = scn.nextInt();
int min = Math.min(x, y);
int gcd = 1;

for(int i = 1; i <= min; i++){
    if(x % i == 0 && y % i == 0){
        gcd = i;
    }
}

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

```
6 public static void main(String[] args) {
7     Scanner scn = new Scanner(System.in);
8
9     int t = scn.nextInt();
10
11     for(int k = 0; k < t; k++){
12         int x = scn.nextInt();
13         int y = scn.nextInt();
14         int min = Math.min(x, y);
15         int gcd = 1;
16
17         for(int i = 1; i <= min; i++){
18             if(x % i == 0 && y % i == 0){
19                 gcd = i;
20             }
21         }
22         System.out.println(gcd);
23     }
24 }
25
26 }
```

```
public static void gcd1(int x, int y){  
    int min = Math.min(x, y);  
    int gcd = 1;  
  
    for(int i = 1; i <= min; i++){  
        if(x % i == 0 && y % i == 0){  
            gcd = i;  
        }  
    }  
    System.out.println(gcd);  
}
```

```
public static int gcd2(int x, int y){  
    int min = Math.min(x, y);  
    int gcd = 1;  
  
    for(int i = 1; i <= min; i++){  
        if(x % i == 0 && y % i == 0){  
            gcd = i;  
        }  
    }  
    return gcd;  
}
```

Prime checker 2

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

eg. 2
12
17

Write a java program to check whether a number is a Prime number or not.

eg. 1 → T
7 → Prime

```
for ( i = 1 ; i <= n ; i++)  
{  
    count factor > 2 → Non-prime  
    else → Prime  
}
```

Prime Number:

↳ which can divisible by 1 & itself only

eg. 7 → 1, 7 → 2
9 → 1, 3, 9 → 3 → Non-prime
6 → 1, 2, 3, 6 → 4
5 → 1, 5 → 2

```
if (factor == 2)  
{  
    Prime  
}  
else {  
    Non-Prime  
}
```

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int t = scn.nextInt();
9         for(int k = 0; k < t; k++){
10             int n = scn.nextInt();
11
12             int count = 0;
13             for(int i = 1; i <= n; i++){
14                 if(n % i == 0){
15                     count++;
16                 }
17             }
18             if(count == 2){
19                 System.out.println("Yes");
20             }else{
21                 System.out.println("No");
22             }
23         }
24     }
25 }
26 }

```

1
7

t=1

k=0

0 < 1 ✓

n = 7

count = 0 / 2

i=5
5 ≤ 7 ✓

7 % 5 == 0

i=1

1 ≤ 7 ✓

7 % 1 == 0 ✓

i=6

7 % 6 == 0

i=3

7 % 3 == 0

i=8
8 ≤ 7 ✗

i=7
7 ≤ 7 ✓

7 % 7 == 0

i=2
2 ≤ 7 ✓

7 % 2 == 0

i=4
4 ≤ 7 ✓

7 % 4 == 0

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int t = scn.nextInt();
9         for(int k = 0; k < t; k++){
10             int n = scn.nextInt();
11
12             int count = 0;
13             for(int i = 1; i <= n; i++){
14                 if(n % i == 0){
15                     count++;
16                 }
17             }
18             if(count <= 2){
19                 System.out.println("Yes");
20             }else{
21                 System.out.println("No");
22             }
23         }
24     }
25 }
26 }
```


HW → ✓ Revise Prime Factorization of a number

Print all factors of a number

eg. $n=8$

1
2
4
8

$n=9$

1
3
9

$n=10$

1
2
5
10

~~logic.~~

```
for ( i
```

→ $\leq n$)

$n \% i == 0$, print (i) .

```
}
```

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         for(int i = 1; i <= n; i++){
10             if(n % i == 0){
11                 System.out.println(i);
12             }
13         }
14     }
15 }

```

1
2
4
8

$$n = 8$$

$$i = 1 \quad 1 \leq 8$$

$$n \% i == 0$$

$$8 \% 1 == 0$$

~~2~~

$$2 \leq 8$$

$$8 \% 2 == 0$$

$$i = 3$$

$$3 \leq 8$$

$$8 \% 3 \neq 0$$

$$i = 4 \quad 4 \leq 8$$

$$8 \% 4 == 0$$

$$i = 5 \quad 5 \leq 8$$

$$8 \% 5 \neq 0$$

$$i = 6 \quad 6 \leq 8$$

$$8 \% 6 \neq 0$$

$$i = 7 \quad 7 \leq 8$$

$$8 \% 7 \neq 0$$

$$i = 8 \quad 8 \leq 8$$

$$8 \% 8 == 0$$

$$i = 9 \quad 9 \leq 8$$

boolean type function.

(, which can true / false.

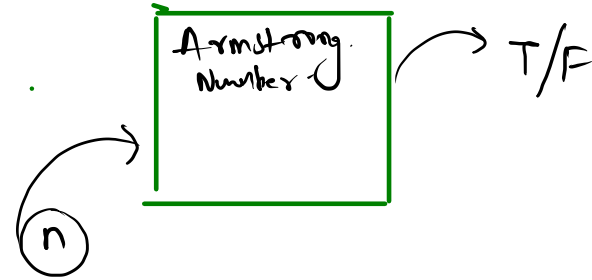
Write a function to check if an Armstrong number or not

Sample Input 0

2 → T
153
234

Sample Output 0

true
false



original Number

1 5 3

1.
$$\text{newNumber} = \{ 3 \times 3 \times 3 + 5 \times 5 \times 5 + 1 \times 1 \times 1 \}$$
$$= 153$$

2.
$$\text{if } (\text{originalNu} == \text{newNu})$$
$$\{$$
$$\quad \text{return } \underline{\text{true.}}$$
$$\}$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static boolean checkArmstrong(int n){
6         int originalNu = n;
7         int newNu = 0;
8         while(n > 0){
9             int d = n % 10;
10            newNu += (d*d*d);
11            n /= 10;
12        }
13        if(newNu == originalNu){
14            return true;
15        }else{
16            return false;
17        }
18    }
19    public static void main(String[] args) {
20        Scanner scn = new Scanner(System.in);
21        int t = scn.nextInt();
22        for(int i = 0; i < t; i++){
23            int n = scn.nextInt();
24            boolean ans = checkArmstrong(n);
25            System.out.println(ans);
26        }
27    }
28 }

```

1
153

t=1

i=0

0 < 1 ✓

n=153

originalNu=153

newN = ~~0~~ ~~7~~ ~~15~~ 2 (153)

153 > 0

d=3

3x3x3

15 > 0

d=5

5x5x5

1 > 0

d=1

1x1x1

User defined function.

$$x=2$$

$$\underline{n=5}$$

$$\frac{1 \times 2 \times 2 \times 2 \times 2 \times 2}{32}$$

$$\Rightarrow 2^n$$

$$prod = 1$$

for (1 \longrightarrow $\leq n$)
{
 $prod \times x = x$
}

Divide n by 2 3 5 and tell steps

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 **visible** 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.

$$n = 30$$

$$\text{Steps} = \cancel{0} \cancel{2} \cancel{5} \textcircled{10}$$

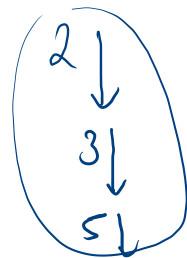
$$30 \rightarrow 15 \rightarrow 5 \rightarrow \textcircled{1}$$

Sample Input 0

30 \rightarrow n
0 \rightarrow steps

Sample Output 0

10 ✓
1 ✓



Sample Input 1

100 ✓
20 ✓

Sample Output 1

34 ✓
1 ✓

$$n = 100 \longrightarrow 50 \longrightarrow 25 \longrightarrow 5 \longrightarrow \downarrow$$

steps = ~~26~~ ~~22~~ ~~21~~ ~~29~~ 34

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int steps = scn.nextInt();
10
11         while(n % 2 == 0){
12             n /= 2;
13             steps += 2;
14         }
15         while(n % 3 == 0){
16             n /= 3;
17             steps += 3;
18         }
19         while(n % 5 == 0){
20             n /= 5;
21             steps += 5;
22         }
23
24         System.out.println(steps);
25         System.out.println(n);
26
27     }
28 }

```

$n = 100 \rightarrow 50 \rightarrow 25 \rightarrow 5 \rightarrow 1$
 $\text{steps} = \cancel{20} \cancel{20} \cancel{24} \cancel{29} \underline{34}$

eg.
 $n = 30$
 $\text{steps} = 0$

$100 \% 2 == 0$ true
 $50 \% 2 == 0$ true
 $25 \% 2 == 0$
 $25 \% 3 == 0$
 $25 \% 5 == 0$

$5 \% 5 == 0$ ✓
 $1 \% 5 == 0$

```

4 public class Solution {
5
6     public static void divide235(int n, int steps){
7         while(n % 2 == 0){
8             n /= 2; ✓
9             steps += 2;
10        }
11        while(n % 3 == 0){
12            n /= 3; ✓
13            steps += 3;
14        }
15        while(n % 5 == 0){
16            n /= 5; ✓
17            steps += 5;
18        }
19
20        System.out.println(steps);
21        System.out.println(n);
22    }
23
24    public static void main(String[] args) {
25        Scanner scn = new Scanner(System.in);
26        int n = scn.nextInt();
27        int steps = scn.nextInt();
28        divide235(n, steps);
29    }
30 }

```

$n = 30 \rightarrow 15 \rightarrow 5 \rightarrow 1$
 $steps = 0 \times 5 / 10$

Arrays } 1st Data Structure.