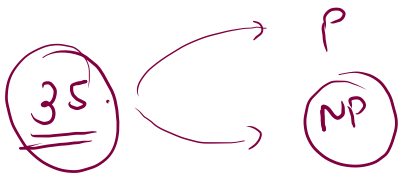
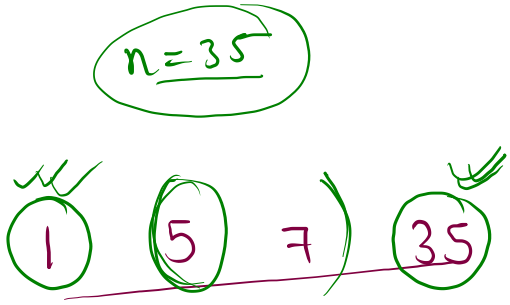
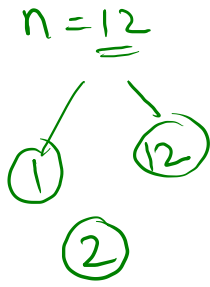
Prime Number







1 X 35

5 X 7

7 X 5

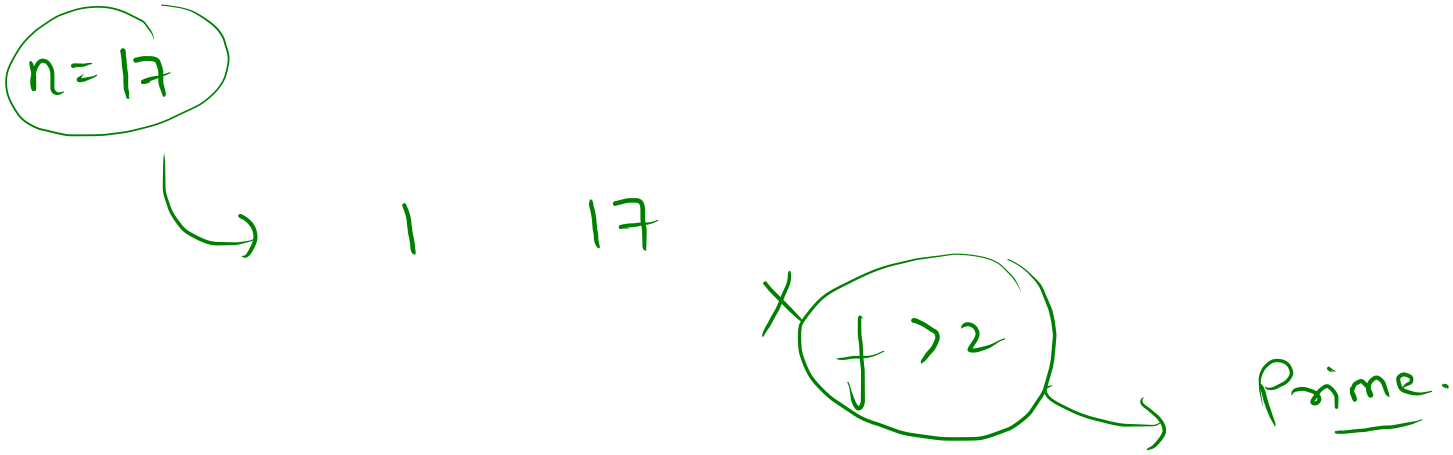
35 X 1

if (f > 2)

{

NP

}


n = 17
1
17
f > 2
Prime.

Prime checker 2

Problem | Submissions | Leaderboard | Discussions

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

Input Format

The first line of input will contain an integer **T** — the number of test cases. The description of **T** test cases follows.

For each test case,

First line contain an integer **N**.

t

n

n = 12

factor = 1 2 3

1 2 3

12 % 1 == 0

12 % 2 == 0

12 % 3 == 0

1 ..... n

factor > 2 → NP

else Prime.

n = 10.

→ 1 2 5 10

1 2 3 4 5 6 7 8 9 10

factor = 0 1 2 3 4

10 % 1 == 0

10 % 2 == 0

10 % 3 == 0 X

10 % 4 == 0 X

10 % 5 == 0

10 % 6 == 0 X

10 % 7 == 0 X

10 % 8 == 0 X

10 % 9 == 0 X

10 % 10 == 0

factor > 2 → NP

else → P

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

        while(t > 0){
            int n = scn.nextInt();

            int factor = 0;
            for(int i = 1; i <= n; i++){
                if(n % i == 0){
                    factor++;
                }
            }
            if(factor > 2){
                System.out.println("No");
            }
            else{
                System.out.println("Yes");
            }
            t--;
        }

    }
}
```

3
12 → No
15 → No
11 → Yes

Prime.

n

more than 2 factor.

1 n

12 → Not Prime.

1 2 12

17

1 17

while
for

t = 1
n = 5

t > 0 ✓

n = 5.

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

public class Solution {

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

        while(t > 0){
            int n = scn.nextInt();

            int factor = 0;
            for(int i = 1; i <= n; i++){
                if(n % i == 0){
                    factor++;
                }
            }
            if(factor > 2){
                System.out.println("No");
            }
            else{
                System.out.println("Yes");
            }
            t--;
        }

    }
}
```

(1)

factor = 0 ✗ (2)

i = 1 ✗ 2 3 4 ✗ 6

2 <= n

2 <= 5 ✓

3 <= 5 ✓

4 <= 5 ✓

5 <= 5 ✓

6 <= 5 ✗

n % i == 0

5 % 1 == 0

5 % 2 == 0 ✗

5 % 3 == 0 ✗

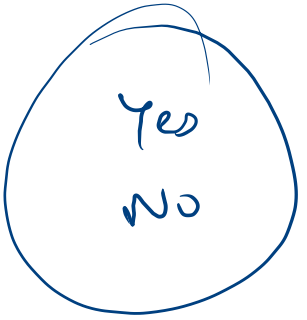
5 % 4 == 0

5 % 5 == 0 ✓

2

17

12



Yes No

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.

Sample Input 0

```
45
```

Sample Output 0

```
3
5
```

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.

n = 12

1 2 3 4 6 12

n = 15.

1 3 5 15

n = 7

1 7

n=12

1 2 3 4 5 6 7 8 9 10 11 12

i

n%i==0

print(i)

12%1==0 ✓

12%2==0

12%3==0 ✓

12%4==0 ✓

1 2 3 4

n=6 ⟶ 1 2 3 6.

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    for(int i = 1; i <= n; i++){
        if(n % i == 0){
            System.out.println(i);
        }
    }
```

1
2
3
6

n=6

i=1 2 3 4 5 6

6%1==0

6%2==0

6%3==0

6%4==0

6%5==0

6%6==0

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 **visible** by **5** and also **increment** the variable steps by **5**, each time you divide the number by **5**.

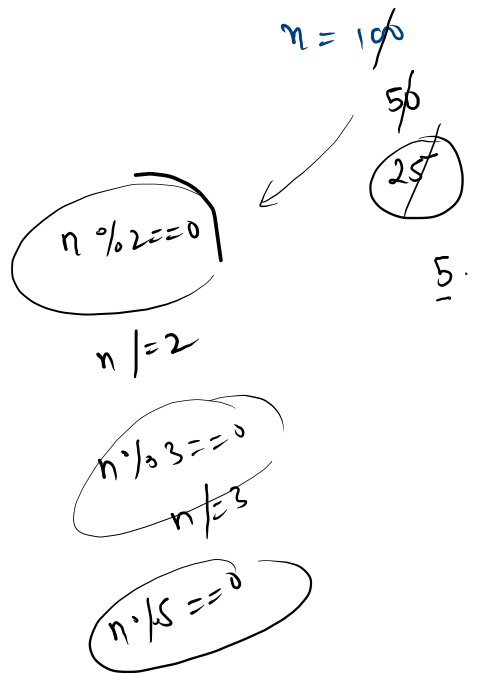
In the end print the value of the variable steps in the first line and final value of number **n** in the second line.

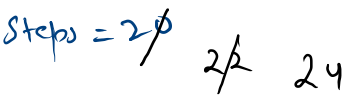
Sample Input 1

```
100 → n
20 → steps
```

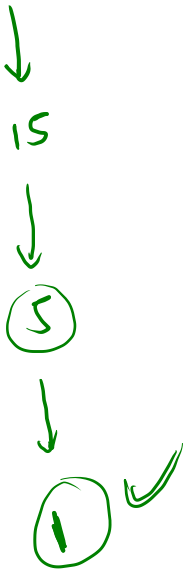
Sample Output 1

```
34
1
```



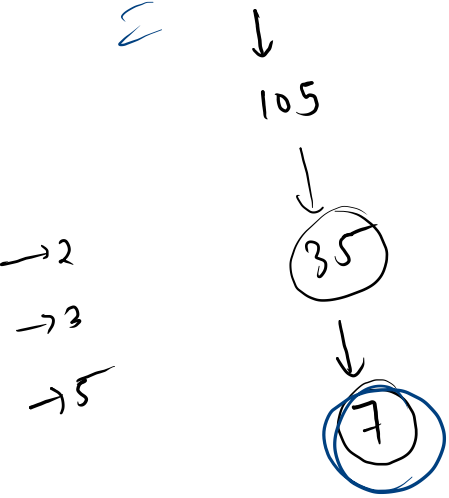


n = 30



steps = ~~0~~ ~~2~~ ~~5~~ 10

n = 210



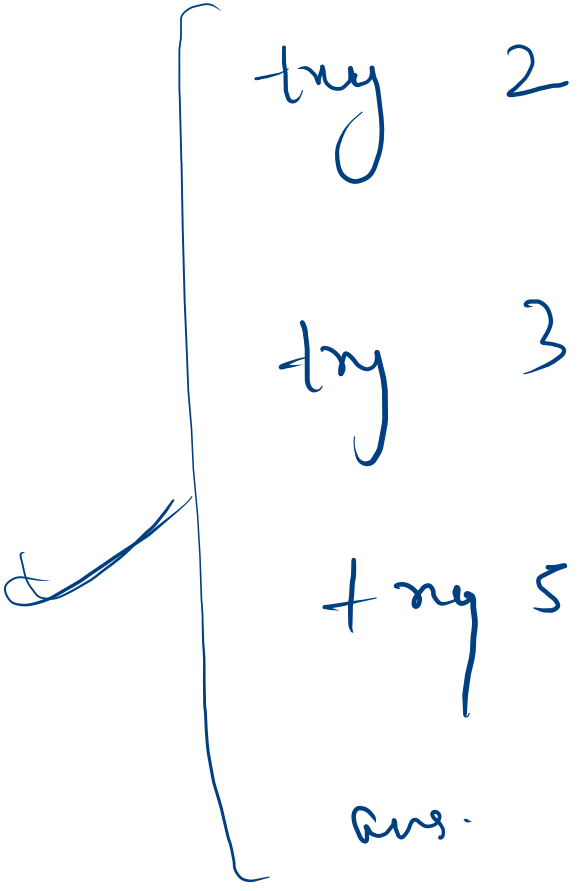
Steps = ~~7~~ ~~9~~ ~~12~~ 17

Ans.

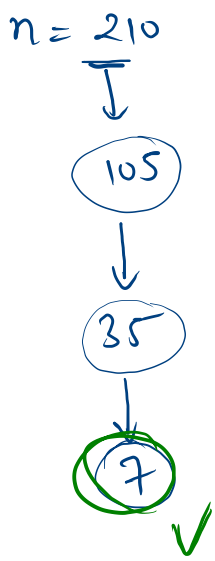
17

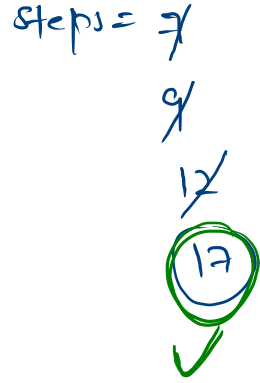
7

Idea.



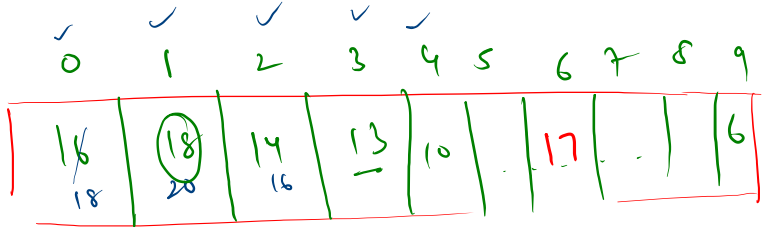
```
8
9           int n = scn.nextInt();
10          int steps = scn.nextInt();
11
12          //try 2
13          while(n % 2 == 0){
14              n /= 2;
15              steps += 2;
16          }
17          //try 3
18          while(n % 3 == 0){
19              n /= 3;
20              steps += 3;
21          }
22          //try 5
23          while(n % 5 == 0){
24              n/= 5;
25              steps += 5;
26          }
27          System.out.println(steps);
28          System.out.println(n);
29      }
30  }
```





int = "aman";

Array. for



→ group of similar kind of thing.

→ continuos elements of same kind.

marks1 ⟶ 16

marks ⟶ 18

marks3 —— 13

marks4 —— 14

marks —— 10

⋮

marks10 —— 6

10 variable.

marks 1 += 2

marks2 += 2

⋮

DS:

1. initialize.

int age = 52; (variable.

2. get

3. put

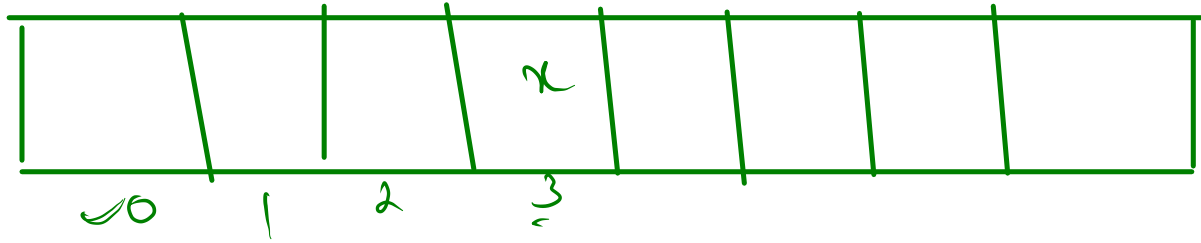
4. travel. / traverse.

Arrays:

Geeks

0 1 2 3 4

Array:



Syntax.

marks arr. (int) ✓

int [ ] nameofArr = new int [size];

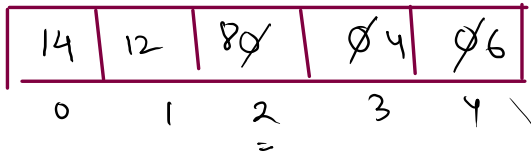
size=5

```
**************************************************
public class Main
{
    public static void main(String[] args) {

        int marks1 = 14;
        int marks2 = 12;
        int marks3 = 8;
        int marks4 = 4;
        int marks5 = 6;

        //initialize the array

        int [] marks = new int[6];
```



1 variable.

marks[i]

Java

default value of int → '0'

marks[0] = 14

marks[1] = 12

age = 12

marks[2] = 8

marks[3] = 4

marks[4] = 6

L = R

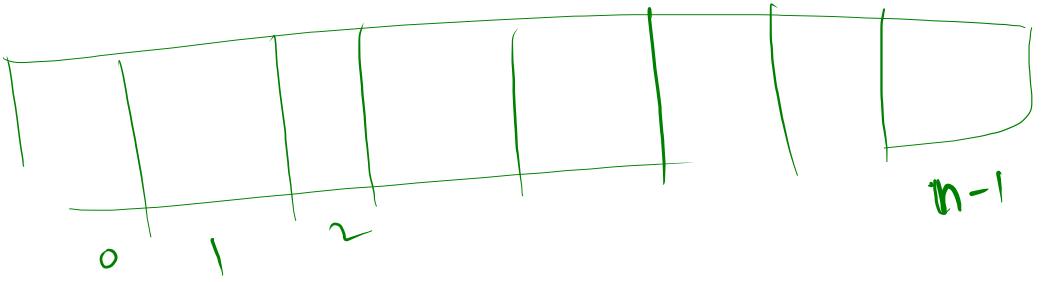
age = 52

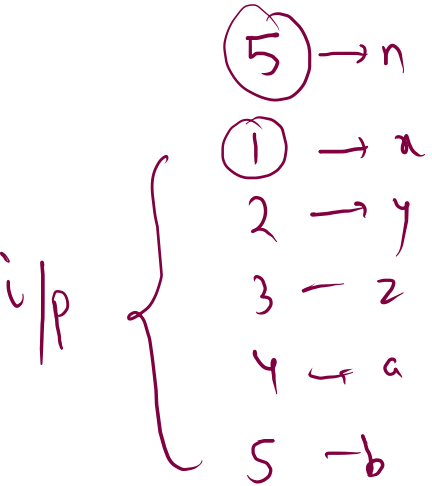
marks[5] = 10

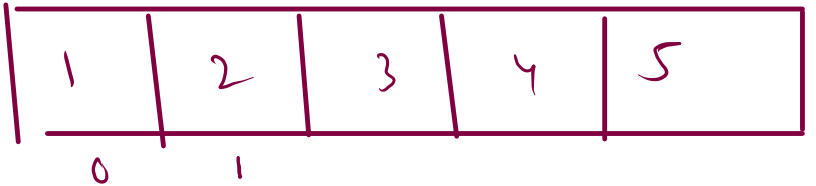
Create array of size n

n = i/p

```
int [] arr = new int [n];
```







x = scn.nextInt()

arr[0] = scn.nextInt();

arr[1] = ;

arr[2] = ;

arr[3] = ...;

arr[n-1] = ;

0 < n-1

Print the array elements linewise

Problem | Submissions | Leaderboard | Discussions

Take n as an integer input. Declare an array of size n that stores value of int data-type.

Then take n integer inputs and store them in the array one by one.

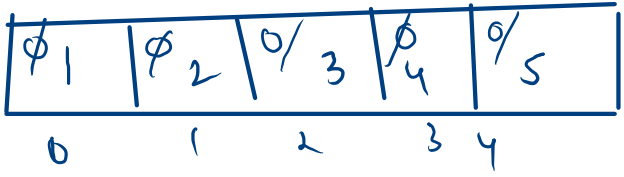
And print each integer in each line.

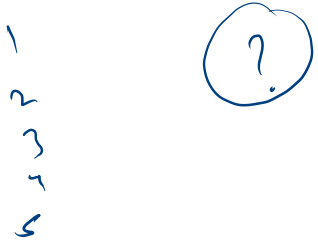
Sample Input 0

```
5
1
2
3
4
5
```

Sample Output 0

```
1
2
3
4
5
```





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

    int [] arr = new int[n];
    //input
    for(int i = 0; i < n; i++){
        arr[i] = scn.nextInt();
    }
    //print
    for(int i = 0; i < n; i++){
        System.out.println(arr[i]);
    }


```

n=5

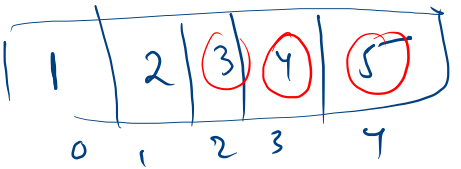
5 ✓
1 ✓
2 ✓
3
4
5

i=0 0<5

arr[0]=1

1<5

arr[1]=2



i=0 ✓ 0<5 ✓ 3<5 ✓

1<5 ✓ 4<5

2<5 ✓ 5<5

1
2
3
4
5