

Doubts

- ① WA Group → Friends
- ② TA Support
- ③ WA mc

② `int cnt = 1` Initialization

`while (cnt ≤ 5)` ② Condition

`System.out.println("Hello")`

`cnt ++`

→ ③ Loop work

④ updation

initialization

`while (condition)` <

 Loop work

 update

`for (initialization ; condition ; update)` <

 Loop work (Task we want to repeat)

```
for (int cnt=1; cnt <= 5; cnt++) <  
    System.out.print("Hello")>
```

cnt=1 ✓

Hello

cnt=2 ✓

Hello

cnt=3 ✓

Hello

4 ✓

Hello

5

Hello

6 ✗

Factors of a number

i is said to be a factor of N if i divides N completely i.e. $N \% i = 0$

Factors of 6 : 1, 2, 3, 6

Factors of 10 : 1, 2, 5, 10

Quiz 3 : Factors of 24 : 1, 2, 3, 4, 6, 8, 12, 24

Factors of $N \rightarrow [1, N]$

Quiz 5: Smallest Prime No. $\rightarrow 2$

Prime no. $\rightarrow 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, \dots$

10:14

Break : used to exit or terminate the nearest enclosing loop prematurely

```
int n = scn.nextInt();
```

```
int countOfFactors = 0
```

```
for (int num = 1; num <= n; num++) {
```

```
    if (n % num == 0) {  
        countOfFactors++;  
    }  
}
```

```
if (countOfFactors == 2) {
```

```
    print("prime")  
}
```

```
else {
```

```
    print("not prime")  
}
```

$n = 24$

Loop will

run 24

times

$cof = 0$

$num = 1$

$cof = 1$

$num = 2$

$cof = 2$

$num = 3$

$cof = 3$

```
int n = scn.nextInt();
int countOfFactors = 0
```

num = 1 3 5
cf = 0 → 1 → 2 → 3

```
for (int num = 1; num ≤ n; num++) {
```

```
    if (n % num == 0) {
        countOfFactors ++
```

```
        if (countOfFactors > 2) {
            break;
        }
    }
}
```

n = 21

```
if (countOfFactors == 2) {
    print("prime")
}
```

```
else {
    print("not prime")
}
```

cf = 0
num = 1
cf = 1
num = 2
cf = 2
num = 3
cf = 3

- ① Avoid unnecessary iterations
- ② used to terminate a loop early based on some condition
- ③ Once a loop terminates, control goes to the immediate next line after the loop.

Nested loop

```
for (int i = 1; i ≤ 5; i++) <
```

```
for (int j=1; j ≤ 7; j++) <
    print (j + " ")
```

```
println (" ")
```

$$i = 1$$

2

3

4

5

6

7

$$i = 2$$

‘

2

3

۴

2

6

7

$i = 3$

2

3

4

1

6

1

$\hat{c} = 4$

2

1

6

1

$$i = 5$$

2

3

4

5

6

1

```
for (int i = 1; i ≤ 5; i++) <
```

```
for (int j=1; j ≤ 7; j++) <
```

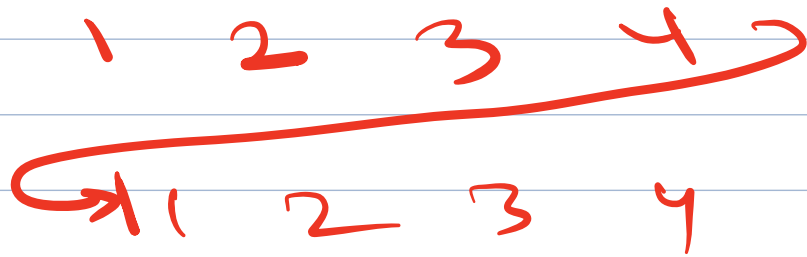
if (j == 5) < break

```
print (j + " ")
```

```
print ln (" ")
```

i = 1

1 2 3 4



i = 2

i = 3

i = 4

i = 5

1 2 3 4

1 2 3 4

1 2 3 4

Continue : skip rest of the current iteration and move to next iteration in loop

Print all odd nos. from 1 to n

```
for (int num = 1 ; num ≤ n ; num++) {  
    if (num % 2 == 0) {  
        continue ;  
    }  
    print (num)  
}
```

n = 7

1 3 5 7

How to solve questions with T test cases?

1. Read the value of T
2. Use a loop to iterate T times
3. Inside loop

3.1 Read the input specific to 1 test case

3.2 Apply logic

3.3 Output of test case

Given T testcases, each testcase will have 1 integer in input, check if it is prime or not.

T

4	6	20	17	1
---	---	----	----	---

4
6
20
17
1


```
int T = scr. nextInt()
for (int cnt=1 ; cnt ≤ T ; cnt++) <
```

```
int n = scr. nextInt() >
```

```
int countOfFactors = 0
```

```
for (int num=1 ; num ≤ n ; num++) <
```

```
    if (n % num == 0) <
        |
        | countOfFactors ++
    >
```

```
if (countOfFactors == 2) <
```

```
    |
    | print (" prime")
```

```
    >
else <
```

```
    |
    | print (" not prime")
    >
```

7

Scope of variable

↓

lifetime of variable

region of a program where that variable is accessible and can be used

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

```
    int a = 10
```

```
    for (int i = 1; i <= 10; i++) {
```

```
        int b = 20
```

```
        // b++
```

```
    }  
    int c = 20
```

```
str num = "1221"  
num2 = "1221"
```

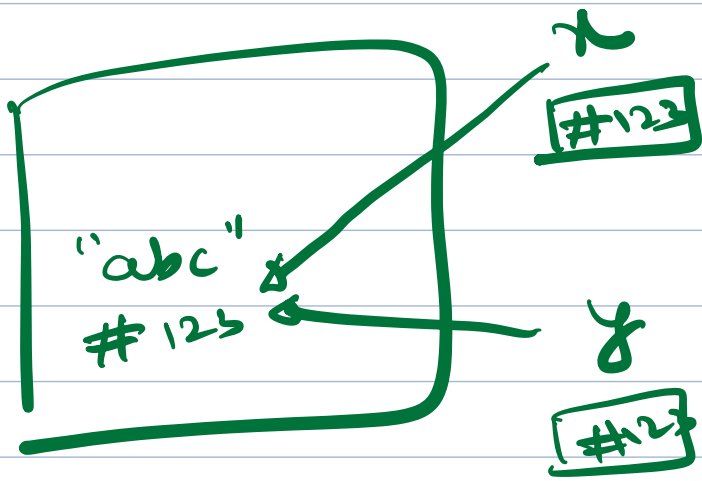
```
a == b
```

a # 123
Aprun

b # 146
sanket

num # 123
1221

num2 # 1221
1221



~~str~~ $x = \text{"abc"}$
 $\text{str } y = \text{"abc"}$