



why do we need loop statement in java?  
loop is a statement in java, it helps the  
programmer to execute a set of instruction  
repeatedly multiple times.

### Types of Loop Statement :-

- while
- do while
- for
- for each / enhanced for

#### while :-

##### Syntax :-

while ( condition )

statement ;

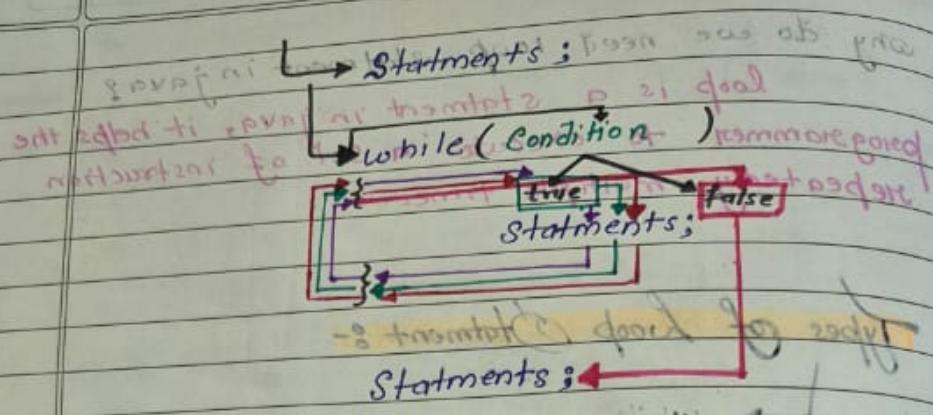
OR

while ( condition )

{  
statement 1 ;  
statement 2 ;  
}

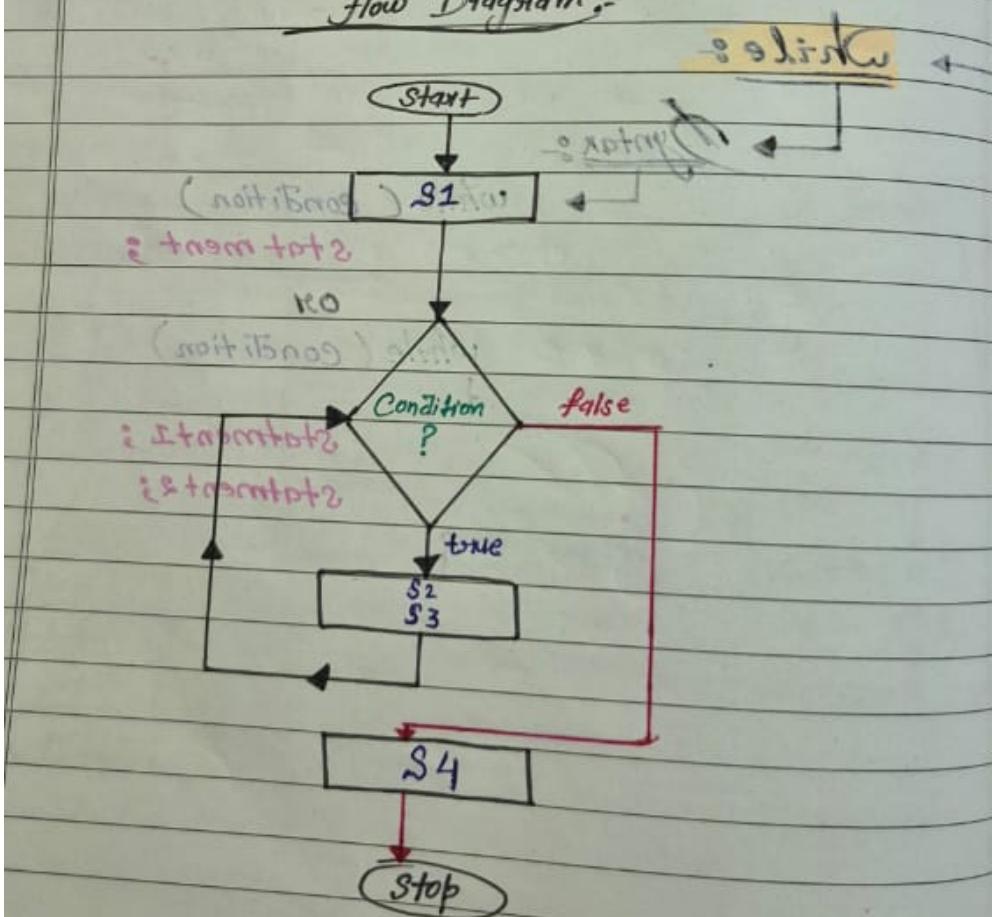
Statement  
S1  
S2

P2



"Here Condition is used to iterate the loop  
if condition is true iterates if false Terminated  
from the loop."

flow Diagram:-



DATE : / /

Write a java program to print \* 5 times.

Count = 0	✓	→ *	print
1	✓	→ *	print
2	✓	→ *	print
3	✓	→ *	print
4	✓	→ *	print
5	X	→ *	terminated from loop.

### Condition :-

Condition	Count $\leq 4$	Count $\leq 4$	Count $< 5$
Count $< 5$	$0 \leq 4 \checkmark$	$0 < 5 \checkmark$	
	$+1$ $1 \leq 4 \checkmark$	$1 < 5 \checkmark$	
	$+1$ $2 \leq 4 \checkmark$	$2 < 5 \checkmark$	
	$+1$ $3 \leq 4 \checkmark$	$3 < 5 \checkmark$	
	$+1$ $4 \leq 4 \checkmark$	$4 < 5 \checkmark$	
	$+1$ $5 \leq 4 X$	$5 < 5 X$	

→ Code :-

int Count (0 == 0) ; lidw

while : ( Condition ) { }

{  
    sopln ("\*");  
    Count++;

2

Example: int Count ;

while ( count < 5 )

2

sop in ("\*");

```
Count = Count +1; // Count += 1; // count++;
```

3

Q7 write a java program to print multiples of 3 upto 10 times.

$$\boxed{3} \times \boxed{1} = \cancel{*} \quad 3$$

$$3 \times 2 = 6$$

$$3 \times 3 = 9$$

六

$$3 \times 10 = 30$$

S N W

$\downarrow$  This part of data is changing by increasing the current value by 1

Code :-

## class Program1

public static void main (String [] args )

*int no = 1;*

while (no <= 10) +n09 +ni

`soplн( 3 * no );` → `3 * no++`

no ++;

\* 51902

++ + 200

107

20.  $\text{Ganu} + \text{lu} + \text{...}$

( $\exists u > v$ )  $\text{Gnu} < \text{Gu}$ ) slides

:((\* \* )) n1d02

$$(t + t_{\text{now}}) = t_{\text{now}}$$

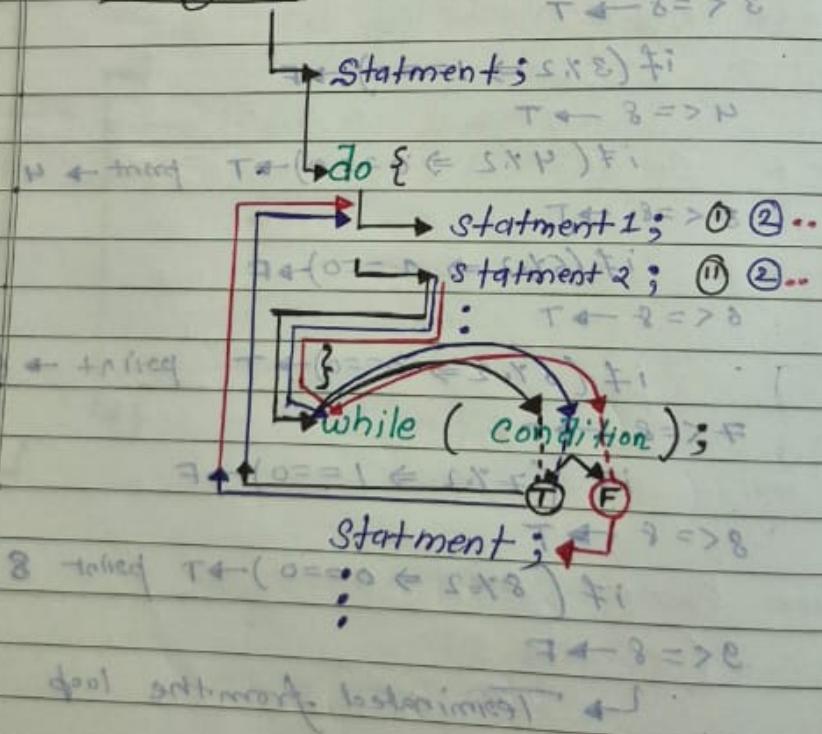


# do while :-  
 it is a loop statement in java, it uses  
 2 keywords  
 → do  
 → while

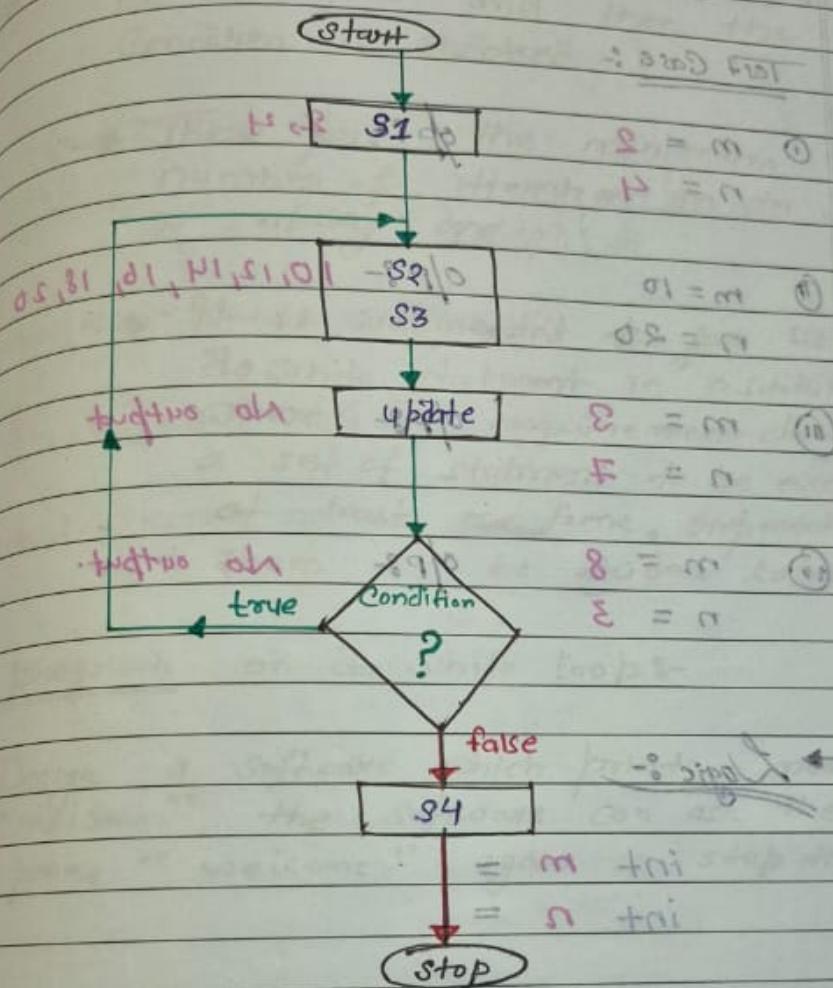
Syntax :-

```
do {  
    Statement1;  
    Statement2;  
    ...  
    }  
while ( condition );
```

work flow :-



## Flow Diagram :-



$$(0 == \Sigma, m) \text{ if}$$

$i (m)$  stop

$i ++ m$

$i (n > m) \text{ stop}$

Q7 write a java program to print even nos between range of m & n ( $m < n$ )

Test Case :-

i)  $m = 2$   
 $n = 4$

O/P :- 2, 4

ii)  $m = 10$   
 $n = 20$

O/P :- 10, 12, 14, 16, 18, 20

iii)  $m = 3$   
 $n = 7$

O/P :- No output

iv)  $m = 8$   
 $n = 3$

O/P :- No output.

→ Logic :-

int

$m =$

int

$n =$

do {

if ( $m \% 2 == 0$ )

System.out.println(m);

$m++;$

while ( $m < n$ );

Note 2 → In do while statement loop block is executed first and then the condition is checked.

→ These form the minimum number of iteration in do while is already done.

→ If you recommend to use do while statement in a situation where requirement demands a set of statement to be executed at least one time, repetition of them will be decided later.

→ program on do while loop :-

Design a software which prints a message "welcome", the software can ask the user print "welcome" again or stop the software.

TC 1 :-

welcome

Do you want to print again

y

welcome

Do you want to print again

TC 2 :-

welcome

Do you want to print again

N

Code 8.1 + method 2 did@OB AB → static  
 Jett next line truly below ex  
 class Program2 extends JPanel

{

    public static void main (String [] args)  
 did@OB AB → Scanner s = new Scanner (System.in);  
 do { System.out.println ("Welcome"); };

    new.nextLine(); → Do you want to print again?  
 did@OB AB → scanner.nextLine(); → Type Y for yes, N for no;  
 scanner.nextInt(); → for int  
 scanner.next(); → char = scanner.next().charAt(0);  
 scanner.nextLine(); → for next line  
 while (ch == 'Y');

-> do { did@OB → for lower case only. } +

    if (ch == 'y' || ch == 'Y') → (ch == 'y' || ch == 'Y')  
 did@OB AB → fast both cases  
 did@OB AB → fast both cases  
 } → fast both cases  
 } → fast both cases

-> SBT  
 welcome  
 did@OB AB → Do you want to print again?  
 Y

-> LST  
 welcome  
 did@OB AB → Do you want to print again?  
 Y  
 welcome  
 did@OB AB → Do you want to print again?

What is loop :-

→ It is a keyword, used as loop statement.

Syntax :-

`for (initialization ; condition ; update)`

↳ Only one  
element is  
present that  
braces  
optional.

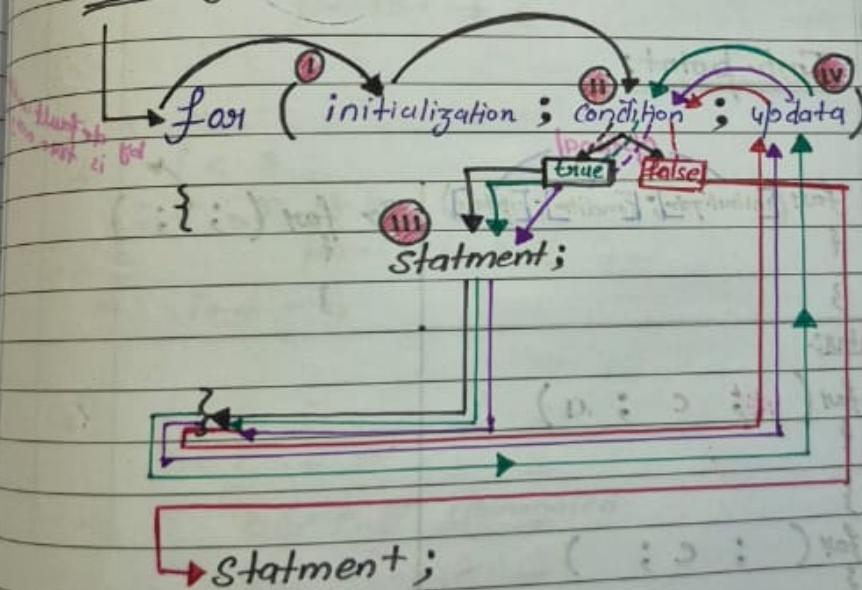
Statement;

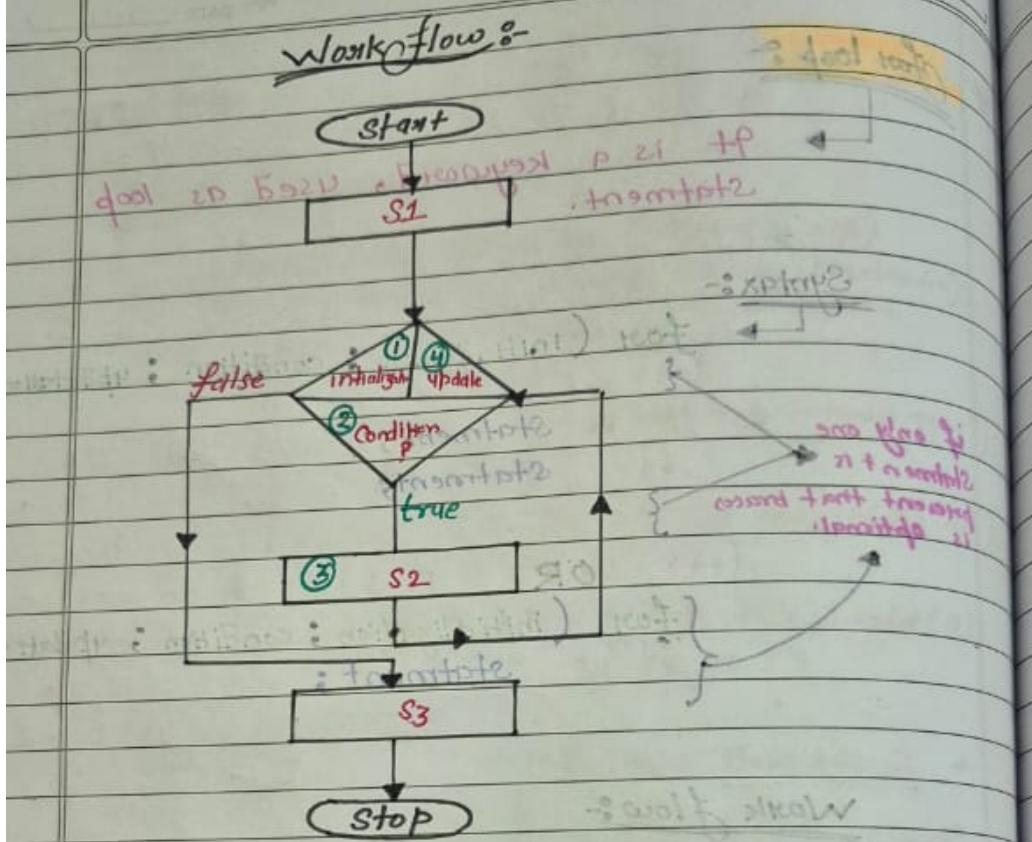
Statement;

OR

`for (initialization ; condition ; update)  
statement ;`

Work flow :-





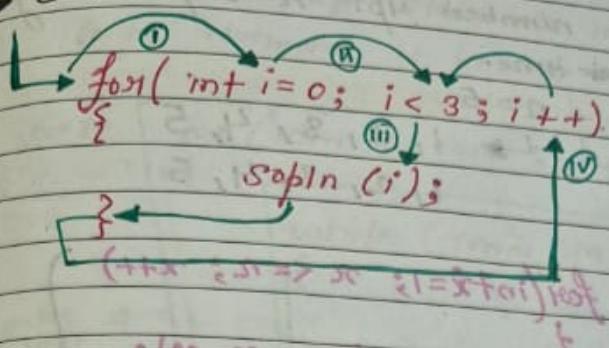
### Imp. point:

for([initialization]; [condition]; [update])  
 {  
 }  
 Syntax:-  
 ↗ for( ; c ; u )  
 {  
 }  
 ↗ for( ; c ; ) : transit 2 ←

optional  
by default condition  
is true every time

## Topic :- Tracing of for loop :-

→ Code :-



$i$

0	$i < 3$	{
	$(i < 3 \rightarrow T \Rightarrow x ; x = x + 1) \text{ not}$	}
	$\text{print:- } i \rightarrow 0$	
	$i++ ( = 1 + 0 ) \text{ not}$	

1       $i < 3$   
 $i < 3 \rightarrow T$   
 $(\text{print:- } i \rightarrow 1)$   
 $i++ = 2$

$(\text{++}x : s = > x ; x = x + 1) \text{ not}$

2       $i < 3$   
 $(2 < 3 \rightarrow T) \text{ not}$   
 $\text{print:- } i \rightarrow 2$   
 $i++ = 3$

3       $i < 3$   
 $3 < 3 \rightarrow F \& \downarrow$   
 $\hookrightarrow \text{Loop is Terminated.}$   
 $2, 1, 0 \downarrow$   
 $2, 1, 0 \downarrow$   
 $2, 1, 0 \downarrow$

## Topic:- Nested Loop :-

a) print numbers upto n separated by comma.  
n time.

~~TC :- n = 58 > i : 0 = i + m ) not~~

$\hookrightarrow 1, 2, 3, 4, 5$   
 $1, 2, 3, 4, 5$

~~for (int i=1; x <= n; x++) {~~

~~sopln (x + ", ");~~

~~X for (int i=1; x <= n; x++) {~~

~~sopln (x + ", ");~~

This is not  
good way becau  
if i want to print  
100 times then  
have to write  
for loop 100 time

$\Sigma > i$

$T \leftarrow \Sigma > i$

~~for (int i=1; i <= n; i++) {~~

$\Sigma = ++i$

~~for (int x=1; x <= n; x++) {~~

$\Sigma > i$

~~sopln (x + ", ");~~

$\Sigma = ++i$

$\Sigma = ++i$

$\hookrightarrow o/p :- n = 5;$

$\Sigma > i$

~~1, 2, 3, 4, 5 > 5~~

~~1, 2, 3, 4, 5~~

~~1, 2, 3, 4, 5~~

~~1, 2, 3, 4, 5~~

~~1, 2, 3, 4, 5~~

Nested loops

**Nested loop**  A loop statement inside another loop statement is known as nested loop.

Ex 8-

for (initit ; cond' ; update)

out  
too

while (condition)

inner  
sep.

$\pm 4.0 \text{ sec} = 18$

卷之三

第十九回

## Topic :- Work flow of nested loop :-

		Iteration		Loop Control	
		i < 2	i = 0	j < 2	j = 0
0	i < 2	$i < 2 \rightarrow T$	$i = 0$	$j < 2 \rightarrow T$	$j = 0$
	( $i < 2 \rightarrow T$ ) when loop starts				for(int i=0; i<2; i++)
j	0	$j < 2$	$i = 0$	$j < 2$	$j = 0$
	$0 < 2 \rightarrow T$				for(int j=0; j<2; j++)
	$i * j = 0 * 0 = 0$ on trai i				
+1	$i < 2$	$i < 2 \rightarrow T$	$i = 1$	$j = 0$	$j = 1$
	$i * j = 0 * 1 = 0$				so print ( $i * j$ );
2	$i < 2$	$i < 2 \rightarrow F$	Break		
	$2 < 2 \rightarrow F$				so print (j);
1	i < 2				
	$i < 2 \rightarrow T$				
j	0	$j < 2$			
	$0 < 2 \rightarrow T$				
	$i * j = 1 * 0 = 0$				
+1	$j < 2$	$i < 2 \rightarrow T$			
	$i * j = 1 * 1 = 1$				
2	$j < 2$	$2 < 2 \rightarrow F$			
2	$i < 2$	$2 < 2 \rightarrow F$			
			Loop terminated.		

Work flow :-  
 inner loop get fully executed for  
 every iteration (cycle) of the outer loop.  
 (do while loop is executed in successive loop)

**Topic & program on loop statements :-**

QP1:- write a java program to print the Factorial  
 of (the given) number.

TC1 :- i/p :- 3 o/p :- 6

TC2 :- i/p :- 5 o/p :- 120

TC3 :- i/p :- 9 o/p :- 362880

$$3! = 3 * 2 * 1$$

$$5! = 5 * 4 * 3 * 2 * 1$$

$$9! = 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1$$

→ do while loop, break, continue, if else

→ Code :-

```
for loop class Factorial
{
    public static void main (String args[])
    {
        int prod = 1;
        int no = 0;
        for (int i = 1; i <= no; i++)
        {
            prod = prod * i;
        }
        System.out.println (prod);
    }
}
```

$$\begin{array}{c} \text{S > i} \\ \text{T < S > i} \\ \text{O = O + S = C * i} \end{array}$$

$$\begin{array}{c} \text{S > i} \\ \text{T < S > i} \\ \text{S = S + i = C * i} \end{array}$$

$$\begin{array}{c} \text{S > i} \\ \text{T < S > i} \end{array}$$

$$\begin{array}{c} \text{S > i} \\ \text{T < S > i} \\ \text{S = S + i = C * i} \\ \text{T < S > i} \end{array}$$

do while loop, break, continue, if else

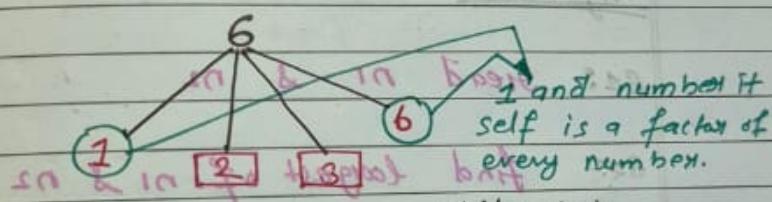
```
int result = 1;  
int i = 2;  
while (i <= n)  
{  
    result = result * i; // This part  
    i++; // This part  
    cout(result);  
}
```

### Problem 2 :-

→ Write a program to find the factors for a given natural numbers.

TG1 :- !/P :- g o/P :- 1,3,9

Tc2 :-  $i/p \approx 15$        $o/p \approx 1, 3, 5, 15$



~~Fay 100 b~~

## while loop

int no = ;    S = 75H int no = ;

for (int i = 2; i < n / 2; i++) { int j = 2;

for ( $i = 1$  to  $n$ ) {  
    **if** ( $i \leq n/2$ ) **do bubbleSort**( $A[1:n/2]$ );  
    **else** **do bubbleSort**( $A[n/2:n]$ );  
}

if ( $100\% i = 0$ ) then  $i = 0$

$$\{o = i \cdot 5 \cup b \wedge o = i\}$$

$\text{so}\beta\ln(i); i = 70$

$\text{so}\beta_1 \cap (\beta_i)_{\leq i} = \emptyset$

- 14 -

20

七

Digitized by srujanika@gmail.com

卷之三

Digitized by srujanika@gmail.com

— 1 —

Digitized by srujanika@gmail.com

Program 4 :-

→ write a program to calculate the GCD (HCF) of two given natural numbers.

→ Euclid's algorithm

$$TC1 := i/p := 15 \quad i/p := 25 \quad i/p := 5$$

$$TC2 := i/p := 6 \quad (i/p := 9) \quad \text{slide} \quad i/p := 3$$

$$+ i = \text{factors of } n_1 = 15 \rightarrow 1, 3, 5, 15$$

$$n_2 = 25 \rightarrow 1, 5, 25$$

$$\therefore (1, 5) \text{ HCF}$$

$$\text{HCF} \rightarrow 5$$

$$\text{factors of } n_1 = 18 \rightarrow 1, 2, 3, 6, 9, 18$$

$$n_2 = 36 \rightarrow 1, 2, 3, 4, 6, 9, 12, 18, 36$$

$$e.g., 1, 2, 3, 6, 9, 18 \rightarrow \text{HCF} \rightarrow 18$$

$$e.g., 1, 2, 3, 6, 9, 12, 18, 36 \rightarrow \text{HCF} \rightarrow 6$$

Algorithm :-

S1 :- read  $n_1$  &  $n_2$

S2 :- find largest of  $n_1$  &  $n_2$

S3 :- initially HCF = 1 ;  $i = n_1 + n_2$

$i = i + n_1$  ( $i \geq n_1 \Rightarrow i = i + n_1$ ) loop

$\Rightarrow$  Find HCF

4.1. check for (from  $i \geq \text{largest}/2$  upto 2)

( $i == n_1 \& n_2$ ) 4.2. if ( $n_1 \& n_2 \& i == 0$   $\&$   $n_1 \& n_2 \& i == 0$ ) {

; ( $i$ ) if { }  $HCF = i$ ; ( $i$ ) if { }  $brea$  { }

S5. :  $i++$  print { }  $HCF$

int n1 = s.nextInt();  
int n2 = s.nextInt();

// find largest

int largest = n1 > n2 ? n1 : n2;

// initialize hcf = 1st

int hcf = 1;

// find common factor from i = largest/2 up to 2

for (int i = largest/2; i >= 2; i--)

i = sum + i

if sum == i { // checks if i is common factor

if (n1 % i == 0 && n2 % i == 0)

{

(& sum % i == 0 == sum % hcf) { i = i;

break;

;(i) } end of if

} } end of for

System.out.println(hcf);

i++;

}

ideal nof

i = sum + i

i = sum + i

if sum == i && sum < sum = i + i

(+i : ; ) nof

(& sum % i == 0 == sum % i) { i

{

;(i) nof2

System.out.println(

{

{

Program 5:

write a java program to find the LCM of two numbers.

for :  $Tc1 < i/p_1 - 15 + i/p_2 - 25 + o/p_1 -$   
 $Tc2 = i/p_1 - 20 \quad i/p_2 - 30 \quad o/p_1 -$

$i = fcd + ns$

while :

not took modulus break;

(--i int num1 = i/p1; --i num2 = i/p2) not

int num2 = ;

int i = num1 > num2 ? num1 : num2;

{ a = 1; s continue (true) so } { }

: i = if ( $i \% num1 == 0 \&& i \% num2 == 0$ )

i = fcd

System.out.println (i);

break; } { }

{ i = fcd } and for

i++; }

}

for loop:

int num1 = ;

int num2 = ;

int i = num1 > num2 ? num1 : num2;

for( ; ; i++)

{ if ( $i \% num1 == 0 \&& i \% num2 == 0$ )

System.out.println (i);

break; }

}