

Lecture-1How to print in Java

`System.out.print ("Hello") ;` → To print  
 ↗  
 ↗  
 ↗  
 ↗  
 ↗

User का Program को use करेगा तो  
 उसे इस line के semi colon के अंदर वो content  
 likha है, user को इसे प्रिंट करेगा।

A programming language → output

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

→ Print line = println

1) `import java.util.*;`

`public class Main {` → Reproducing IDE में class का नाम  
`public static void main (String [] args)` → Main में पढ़ते होगा।  
 ↗ where M is capital  
 ↗ write inside main initially

`System.out.print ("Hello world") ;`

`System.out.print ("Hello world") ;`

}

Output →

Hello worldHello world

जो output 2 alag alag line है जोकि एवं  
 लिखने में सहायता की

\* To print next message in next line

PAGE NO.

DATE

- 2) System.out.println ("Hello world");  
↓  
Print line message को Print करने के लिए इसका enter मारकर ही, अपला message next line में print होगा।  
System.out.print ("Hello World");  
↓  
Big/Capital letter small letters  
→ Output → HelloWorld  
Hello world

3) Next line में print करने के लिए इसका print का शी use कर सकते हैं।

- 1st line System.out.print ("Hello world \n");  
↓  
Back slash n,  
इसमें ज्ञाता है या निससे enter लग जाता है;  
2nd line System.out.println ("Hello world");  
→ Output → HelloWorld. It makes any difference  
Hello world. in output because  
it presses enter & hence the very next  
message will be affected by it.

Ques. 1.

Print the following pattern → Right Angle A

```

    *
   * *
  * * *
 * * * *
* * * * *

```

public static void main (String [ ] args)

Way 1

```

System.out.println ("*");
System.out.println ("**");
System.out.println ("***");
System.out.println ("*** *");
System.out.println ("*** * *");
System.out.println ("*** * * *");

```

Output

अब इसे कैसे print करेंगे तो

```

*
* *
* * *
* * * *
* * * * *

```

Way 2

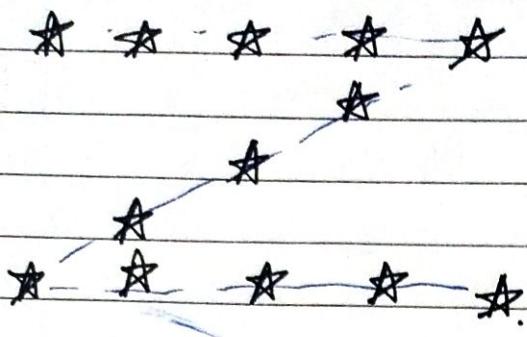
```

System.out.println ("* \n"
                    +"* * \n"
                    +"* * * \n"
                    +"* * * * \n"
                    +"* * * * * ");

```

→ It does its work  
of pressing  
an enter  
key

Ques. 2. Print the following pattern. → 2.



## 5 Basic Skill in a language.

- 1. Print or Output → How to print something
- 2. Taking Input. → How to take input from user
- 3. Variables. → How to store information in Computer memory.
- 4. Conditionals → How to take different actions based on different conditions.
- 5. Loops → How to do something repeatedly.

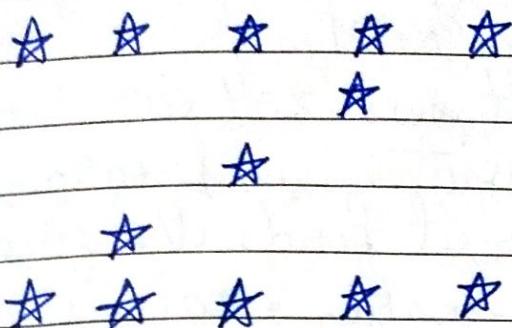
→ Believing & not to go in detail  
Abstraction: It is necessary to know when to go in details and not to go in details.

public static void main()

Way 1  
using System.out.println (" \* \* \* \* \* ");  
println  
System.out.println (" \* \* \* \* \* ");  
System.out.println (" \* \* \* \* \* ");

Ar. Space

Way 2  
using System.out.println (" \*\*\*\* \*\n \* \*\n \*\*\*\* ");

Output

We have learnt the first skill of a language.  
ie To give Output / print.

1. → Print or Output

2. → Variables

3. → Conditionals

4. → Loops

5. → Taking Input

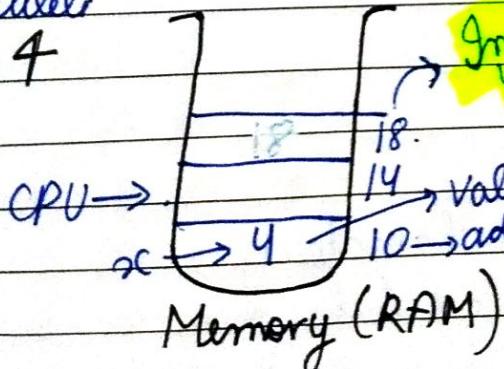
5 Skills of a Language.

2nd Skill → Variables

Hard Disk keeps the data permanent even if Computer

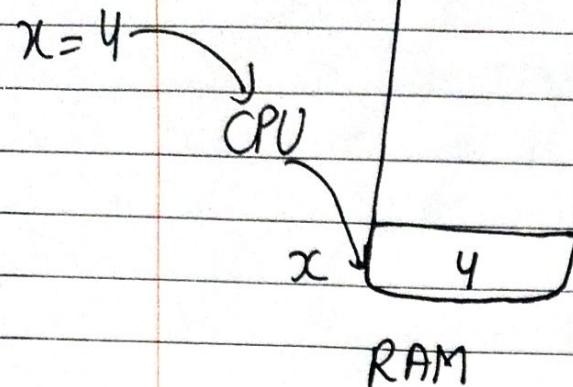
Variables

$$x = 4$$



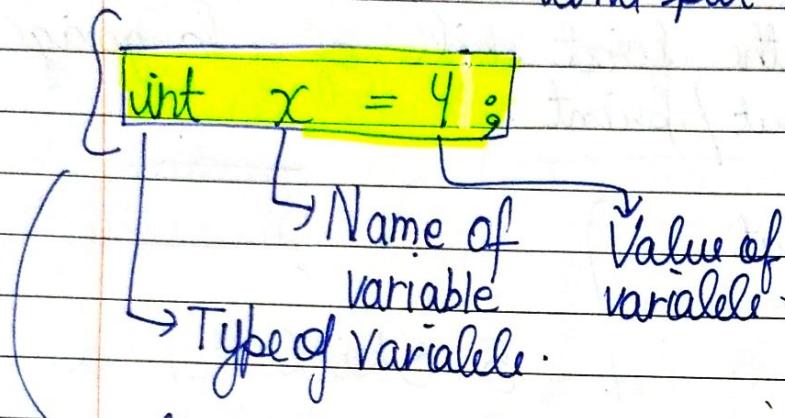
In Java we need not worry about address. But, the data on RAM will go.

But in C++ we need to worry about address.



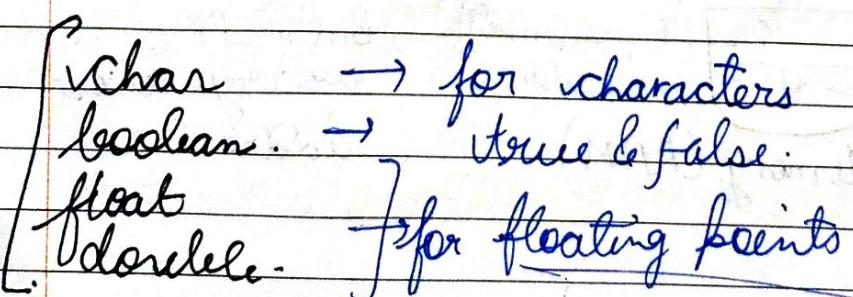
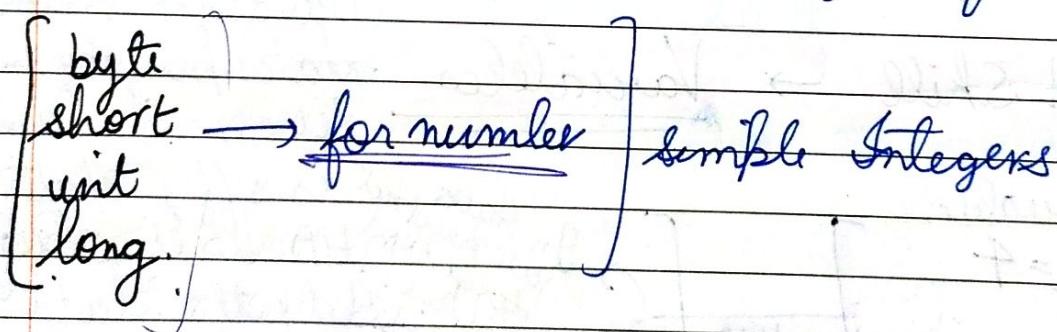
CPU has one eye on program & another on RAM.

It sees  $x=4$  is written in program, so it finds the first empty location & calls it  $x$  and put 4 unto it.



for Java, computer/ compiler is not able to decipher that  $x$  is a variable of type integer.

`int` → Most popular datatype of number



CPV / System / Computer

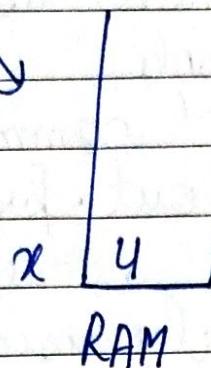
keeps one eye on program

and another eye on RAM

int x = 4;

System.out.println(x);

System.out.println("x");



It will print x

as it is.

'Verbatim' means as it is.

Console

Output → 4.

CW.

1. How to Print in Java
2. How to print Z
3. ||| Pattern of Star
4. What are Variables & How to print

Skill 1.

Ques.

Skill 2

HW

In Getting Started Module

1. How to Print in Java
2. How to print in Java.
3. Print Z.
4. Variables
5. Variables in Java
6. Variables in Java.

Video

Article

Ques.

Video

Article

MCQ.

HW

## Article 1 On

PAGE NO.

DATE

### How to Print in Java

1. `System.out.print` & `System.out.println` are 2 commands used in Java for printing output to console.
2. '`System.out.print`' command prints to the console while "`System.out.println`" prints & then moves to next line as well.
3. '\n' can also be used to move to next line.
4. `System.out.println` does nothing more than printing the message & then printing '\n' after that.

### FACTS

1. Printing numbers does not requires double quoted commas. Try this & find out yourself.
2. `System.out.print("10")` & `System.out.print(10)` will have the same output.

Q.3. Print the following pattern → Equilateral L.

```

      *-----*
      *--* * * *
      *-* * * * *
      *-*-* * * * *
      *-*-*-* * * *
  
```

$n=1$	Space = 4	Star = 1
2	3	3
3	2	5
4	1	7
5	0	9

public static void main {

```

    System.out.println("-----*");
    System.out.println(" *-* * * *");
    System.out.println(" *-*-* * * *");
    System.out.println(" *-*-*-* * * *");
    System.out.println(" *-*-*-*-* * * *");
  
```

## Variables

(Video notes)

PAGE NO.

DATE

"My name" = String  
32 → Integer  
100.50 → float

Java में किसी भी data का variable करने से पहले, हमें कानून पढ़ना है कि उस data का datatype है। जैसे data store करने की बात आज्ञा है, तो हम datatype भी लगाएं।

int x = 15;

int y = 10;

int sum = x + y;

System.out.println("sum"); sum: x+y.

y 10.

x 15

System.out.println("Sum of "+ sum); RAM.

Sum

x +

" and " +

y +

" is ".t.

sum);

int product = x \* y;

Product System.out.println("Product of " + x +  
" and " + y +  
" is " + product);

int v1 = x / y;

int v2 = y / x;

int v3 = x % y;

15/10=1. / gives quotient

System.out.println(v1);

System.out.println(v2);

System.out.println(v3);

Divide

&

Quotient

$$\frac{xy}{x+y}$$

$\rightarrow \text{int exp} = x * y / x + y ;$   
 $\text{System.out.println(exp);}$

$$\boxed{\star / \%} > \boxed{+}$$

equal priority      equal priority

If equal then  
start from left to right.

$$\begin{aligned} \text{int } x &= 10 \\ \text{int } y &= 15 \end{aligned}$$

$$\text{int exp} = x * y / x + y ;$$

$$\text{Without brackets} \quad = 10 * 15 / 15 + 10$$

$$= 150 / 15 + 10$$

$$10 + 10 = 20$$

$$\text{With brackets} \quad = (10 * 15) / (15 + 10)$$

$$= 150 / 25 = 150 / 25 = 6.$$

## Article - 2. On Variables

PAGE NO.

DATE

1. To declare a variable, you need to know its data type as well.
2. When you declare a variable, say `int x = 10;` a location in Computer's RAM is named `x` & 10 is stored in it.
3. / gives quotient & % gives remainder.
4. There is no BODMAS.  
`*`, `/`, `%` have equal priority & higher than `+` & `-` which are equal to each other in the priority criteria.
5. Among operators with equal priority, generally the one on the left side is executed first.
6. Use parenthesis to over-ride order of execution of operators.

### FACTS

1. There is no operator for exponentiation.  $2^3$  is not 8. This is XOR operator. We'll do it later.
2. To do exponentiation, use Math·pow function.

### Exercise

1. Try this code in the codebook & meditate on it.  
`System.out.println(Math.pow(2,3));` → gives 8
2. Try this code in codebook & meditate on it.  
`int exp = (int) Math·pow(2,3);` → gives 8  
`System.out.println(exp);`
3. Compare the output of above 2 & think why the above code has a `(int)` before `Math·pow` & what it does?

# Swapping 2 no without using 3rd Variable

PAGE NO			
DATE			

MCQ-1

first LHS  
goes

What is the output of the following code?

first RHS  
calculated

then answer  
is assigned

in RHS

```
int a=10;  
int b=20;
```

```
a=a+b;  
b=a-b;
```

```
a=a-b;
```

System.out.println(a+" "+b);

→ a=10      b=20.

a=30      b=10

a=20.

a=20, b=10;

a	20	20	20
b	20	10	10
a	10	30	20

first RHS evaluated

answer is  
calculated, then

answer is  
assigned to LHS.

This is the way of  
Swapping 2 variable value  
without using any 3rd Variable

## Lecture - 2] getting started with

5 Skills for any language

Output ✓ Done

Input

Variables ✓ Done

Conditionals

Repetitions

Yesterday, we learnt output & variables in Java.  
Today, we would learn about :

{ Input  
Conditionals  
Repetitions

### SKILL 3rd Conditionals

→ If, Else .

Skill

- (i) if
- (ii) else .

L = Line

C = Conditional .

1) → If

✓ — L<sub>1</sub>  
✓ — L<sub>2</sub> .

if (c) {  
— L<sub>3</sub>

— L<sub>4</sub>

} else {

✓ — L<sub>5</sub>

✓ — L<sub>6</sub>

g .

✓ — L<sub>7</sub>

✓ — L<sub>8</sub> .

If the if condition fails then the CPU enters the else block & run L<sub>5</sub>, L<sub>6</sub> .

These lines will run where we have tick if the if condition is not true

# Way - I Using If-Else ↳ for Conditionals

PAGE NO.	
DATE	

## Q. 1. Grading System.

1. We are given marks of a student
2. Display a message based on following rules-
  - 2.1. for marks above 90, print excellent.
  - 2.2. for marks above 80, less than 90, print good
  - 2.3. " " " 70, " " 80, " fair
  - 2.4. " " " 60, " " equal to 70, print meets expectations
  - 2.5. " " " than equal to 60, print below par.

```
if (marks > 90)
    System.out.println ("excellent");
}
else
    if (marks > 80 && marks <= 90)
        System.out.println ("good");
    else
        if (marks > 70 && marks <= 80)
            System.out.println ("fair");
        else if (marks > 60 && marks <= 70)
            System.out.println ("meets
expectations");
        else if (marks <= 60)
            System.out.println ("below
par"); }
```

```
if (marks > 90)
{
    System.out.println ("excellent");
}
else {
    if (marks > 80)
        System.out.println ("good");
    else {
        if (marks > 70)
            System.out.println ("fair");
        else
            if (marks > 60)
                System.out.println ("meets expectation");
            else
                System.out.println ("below par");
    }
}
```

Way I Using If - Else .

PAGE NO		
DATE		

Q.2 Tells if no. is even/odd

Even  $\rightarrow$  No. divisible by 2.

Odd  $\rightarrow$  no. not divisible by 2.

### Operators

and

or

$+$   $\rightarrow$  add

$-$  Subtract

$*$  multiplying

$/$  slash  $\rightarrow$  divide  $\rightarrow$  quotient

$\%$  modulus  $\rightarrow$  remainder

Divide  $\rightarrow$  Quotient  $16/3 \rightarrow 5$

Modulus  $\rightarrow$  Remainder  $16 \% 3 \rightarrow 1$ .

$$2 \sqrt{21} \rightarrow \text{Odd}$$

20  
1  $\rightarrow$  remainder  $\neq 0$   
 $\therefore$  Odd.

$$2 \sqrt{20} \rightarrow \text{Even}$$

20  
0  $\rightarrow$  remainder = 0.  
 $\therefore$  Even

```
int marks = Scan.nextInt();
```

```
if (marks % 2 == 0)
```

```
{ System.out.println ("Even");
```

```
}
```

```
{ System.out.println ("Odd");
```

```
}
```

If - Else → Simple Conditions

Q.3. Tells if no. is  $3k$  /  $3k+1$  /  $3k+2$ .

$$3k \rightarrow 18, 12$$

$$3k+1 \rightarrow 19, 7$$

$$3k+2 \rightarrow 20, 14$$

$18 \rightarrow 3 \cdot 6$	$\rightarrow 3k$
$12 \rightarrow 3 \cdot 4$	$\rightarrow 3k$
$19 \rightarrow 3 \cdot 6 + 1$	$\rightarrow 3k+1$
$7 \rightarrow 3 \cdot 2 + 1$	$\rightarrow 3k+1$
$20 \rightarrow 3 \cdot 6 + 2$	$\rightarrow 3k+2$

when divided  
by 3, the  
remainders  
can only be  
3 no. or either  
0, or 1 or 2.

int marks = scn.nextInt();

if (marks % 3 == 0)

{ System.out.println ("3K"); }

}

else

{ if (marks % 3 == 1.)

{ System.out.println ("3K+1"); }

}

System.out.println ("3K+2");

}

}

$3K | 3K+1 | 3K+2$

If-Else

if ( $m \% 3 == 0$ )

{  
3K .

else

{

if ( $m \% 3 == 1$ )

{  
3K+1 .

else

{  
3K+2 .

}

}

If-Else-If

If-Else-If

if ( $m \% 3 == 0$ )

{  
3K .

}

else if ( $m \% 3 == 1$ )

{  
3K+1 .

}

else

{  
3K+2 .

}

Commented Code never runs

Single line comment  $\rightarrow //$

Many //  $\rightarrow /*$  \*/

Way II  $3k$  |  $3k+1$  |  $3k+2$   
Using If - else-If

PAGE NO.

DATE

if (marks / 3 == 0)

{  
    System.out.println ("3k");

}

else if (marks / 3 == 1)

{

    System.out.println ("3k+1");

}

else

{

    System.out.println ("3k+2");

}

## My Grading System Using If-Else-If

if (marks > 90)

{

    System.out.println ("excellent");

}

    else if (marks > 80)

{

        System.out.println ("good.");

}

    else if (marks > 70)

{

        System.out.println ("fair.");

}

    else if (marks > 60)

{

        System.out.println ("meets expectations.");

}

else

{

    System.out.println ("below par.");

}

## 4th Skill of a language → Loops

PAGE NO.  
DATE

→ White loop

↳ Repetition

L1  
L2  
while (c)  
{ L3  
 L4  
 }  
L5

L = Line  
C = Condition

Built 1 to 5

Console

unit i = 1;  
while (i <= 5)  
{  
 system.out.println(i);  
 i++;  
}  
System.out.println("done");

1  
2  
3  
4  
5  
done.

RAM

## 5th) Skill Input

PAGE NO.

DATE

Taking something in Input from keyboard  
& then putting the value in RAM.  
variable in

Scanner scn = new Scanner (System.in); To take input  
Variable of type Scanner.

Scanner is a used to read  
from keyboard

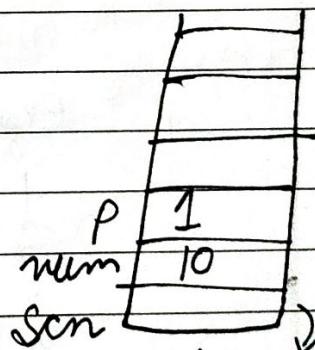
scn  
RAM

```
Scanner scn = new Scanner (System.in);
int num = scn.nextInt();
System.out.println (" You entered " + num);
```

Print num | to n & take n input

```
Scanner scn = new Scanner (System.in);
int num = scn.nextInt();
int p = 1;
while (p <= num)
```

```
    System.out.println (p);
    p++;
}
```



```
System.out.println ("done");
```

Stack scanner  
formed  
on stack

Input	Output
3	1
	2
	3
	done.

~~HW~~

→ Revise & do all these programs

Q1 → Print odd even till n.

~~HW~~

Ques. Print Odd even till n where n is taken as input from user.

```
import java.util.Scanner;  
public class printoddorevenTillN {  
    public static void main (String args[]) {  
        Scanner scan = new Scanner (System.in);  
        int n = scan.nextInt();  
        int x = 1;  
        while (x <= n) {  
            if (x % 2 == 0) {  
                System.out.println (x + " is even");  
            } else {  
                System.out.println (x + " is odd");  
            }  
            x++;  
        }  
    }  
}
```

Lec-3.

## Lecture - 3

PAGE NO.

DATE

4 Sept 2021

Lec-2 Homework → Print Odd Even till n.

↓ Solution by Sir

Q-Take Input n from user & print counting up to even till n

→ import java.util.\*; → Recoding editor में 2<sup>nd</sup> line को delete ना करना और class name

public class Main {

दैवी Main होगा।

public static void main  
(String[] args)

↑ Capital M

→ codechef QR शीर्षी टाइटल होता है।

{ Scanner scn = new Scanner(System.in); }  
int n = scn.nextInt(); } इस line पर

Scanner  
type of object  
variable के समान हैं  
इसे input  
read कर  
ram में assign  
करने के काम  
आता है।

int p = 1;

while (p <= n)

terminal / console को दरवाजा

अब वहाँ सही num पड़ा है।

{ int rem = p % 2;

अब वहाँ सही उत्तर का नाम के

if (rem == 0)

variables में डाल दिया

जाएगा और ram में store

{ System.out.println(p + " is even"); }

दर दिया जाएगा।

} else

{ System.out.println(p + " is odd"); }

{ } Input / Output

7

1 is odd

2 is even

3 is odd

x	81
p	72134567
num	7
Scanner	scn

## Q.1 Is a number prime

### Way 1

तो कहि भी no. prime तब होता है जो वो 1 और स्वयं से ही divisible हो बस, और किसी भी no. से अगर divisible हो जाए 1 और n को छोड़के, तो वो no. non-prime है। A Prime no. should have only 2 factors till n  
 $\therefore 2$  is a Prime no.

So, Way 1 में हम क्या करेंगे कि क्या वो no. n,

Way 2] (2) से लैकर (n-1) तक किसी भी no. से divide हो पा रहा है नहीं

So Prime no. बस वही होते हैं जो 1 से n तक केवल 2 से भी divide हो पाया।  $\rightarrow$  2<sup>nd</sup> Way 1 है check करने के लिए

अगर 2 - (n-1) तक सक बार भी divide ना हो तो no. होगा prime.

Checking 11 by this Way 1.

$2 \times$	$5 \times$	$8 \times$
$3 \times$	$6 \times$	$9 \times$
$4 \times$	$7 \times$	$10 \times$

2 से (n-1) तक कई भी divide नहीं कर पाया।

$\therefore 11$  is Prime

**Way 3.**

(2) से  $(n-1)$  तक check करने की जरूरत ही नहीं है।

→ Check 50 का फॉन-फॉनसे no. से divide होता है।

50

1 ✓

2 ✓

5 ✓

10 ✓

25 ✓

So 25 के बाद 50 तक से divide  
नहीं होता except 50 itself

→ Check 6 का फॉन-फॉनसे no. से divide होता है?

6

1 ✓

2 ✓

3 ✓

तो 6 और 3 के बाद निचले से divide  
नहीं होता except 6.

तो (2) से  $(n-1)$  तक जाने की ओर factor of n check  
करने की जरूरत नहीं है।

$\frac{n}{2}$  तक ही factor मिलसकता है जो का,  
उसके बाद तो n ही होता है।

→ Even no. can never be prime except 2

/ → gives quotient

% → gives remainder

→ तो हमें वस 2 से  $\frac{x}{2}$  तक ही check करने की जरूरत  
है।

$$9 - \frac{x}{2}$$

## Way 4

हमें 2 वे  $\binom{n}{2}$  तक check करने की जरूरत नहीं है।

\* Check factorization of 36.

Unique factors  
जूँके वाले नहीं  
मिलते।  
जिसे भी

$1 \times 36 = 36$	$36 \times 1 = 36$
$2 \times 18 = 36$	$18 \times 2 = 36$
$3 \times 12 = 36$	Mirror Image $12 \times 3 = 36$
$4 \times 9 = 36$	$9 \times 4 = 36$
$6 \times 6 = 36$	

unique factors मिलने होते हैं वे जूँके तक मिल जाते हैं।

\* Check factorization of 25.

$$\begin{array}{l} 1 \times 25 \\ 5 \times 5 \\ 25 \times 1 \end{array} \xrightarrow{\text{Square root of around mirror image}} \text{around mirror image}$$

\* Check factorization of 64.

जो पहला Quotient हो, mirror image of divisor होगा।

$$\begin{array}{l} 1 \times 64 \\ 2 \times 32 \\ 4 \times 16 \end{array} \xrightarrow{\text{Mirror Image}} \begin{array}{l} 64 \times 1 \\ 32 \times 2 \\ 16 \times 4 \end{array}$$

$8 \times 8$

Square root of around mirror image होगा।

p, q फिर Product होगा जिसमें p & q, both cannot be greater than 5.

\* Check factorization of 35 =  $\sqrt{35} = 5 \cdot$  something  
 $2 \times$   
 $3 \times$   
 $4 \times$   
 $5 \checkmark$

→ तो 5 तक ही इस check करें। दोहरा divide करता non prime  
 और  $q$  दोनों  $\sqrt{n}$  से बड़े नहीं  
 हो सकते। या तो  $p$  और  
 $q$ ,  $\sqrt{n}$  के equal हो सकते  
 या  $p$  वा  $q$  दोहरा  $\sqrt{n}$  से  
 हो जाएगा। या vice versa.

$n = pq$  →  $p < \sqrt{n}$  वा  $q < \sqrt{n}$ .

$$p \text{ & } q \neq \sqrt{n}$$

### \* Prime Check करने के ways

→ Way 1 [1 to  $x$ ] → 1 to  $x$  तक divide करें।  
 और वस्तु 2 से  $x$  तक divide होता है।  
 मरम्मत prime है।

→ Way 2 → [2 to  $n-1$ ] → 2 से  $x-1$  तक divide करें।  
 करें तो देखो और कोई भी divide न  
 हो तो prime है।

→ Way 3 → (2 to  $\frac{x}{2}$ ) → 2 से  $\frac{x}{2}$  तक ही check करें।  
 factors 2 से  $\frac{x}{2}$  तक कोई factor  
 factor नहीं मिला except  $x$ .  
 तो अगर 2 से  $\frac{x}{2}$  तक कोई factor  
 नहीं मिला तो  $x$  मरम्मत prime है।

→ Way 4 (2 to  $\sqrt{x}$ ) → 2 से  $\sqrt{x}$  तक ही check करें।  
 योकि  $\sqrt{x}$  के बाद कोई ज्ञाता no. divide  
 नहीं करता। अगर पहले 36 को 4 ने divide  
 किया है। Quotient 9 लगाके तो बादमें  $\sqrt{36}$  के  
 9 divisor होंगा, 4 quotient भी होंगा।  
 नया no. नहीं मिलेगा। 2 से  $\sqrt{x}$  तक किसी भी  
 divide नहीं हुआ न हो तो prime है।

Code for

## Check Prime

प्रिमे छोड़ करके देखना है।

PAGE NO.

DATE

$$\begin{aligned}
 & (\text{divisor})^2 \leq n \leftarrow \text{Square} \\
 & \text{root divisor} \leq \sqrt{n} \\
 & (\text{divisor} * \text{divisor}) \leq n
 \end{aligned}$$

→ int dive = 2 ;

→ boolean flag = true ; // assume कि यह prime है।

→ while (div \* div <= n)

{  
    int rem = n % dive ;

    if (rem == 0)

    {  
        flag = false ;

        break ; → break loop को तुरंत छोच में दी जाती है।

    dive ++ ;

}

[n=35]

→ if (flag == true)

{  
    System.out.println  
        ("x + "is prime");

else

{  
    System.out.println  
        ("x + "is not prime");

or x=30

x 35

dive 2 3 4 5

flag true/false

35 is not prime

dive 1

n=53

flag  
true  
x 3  
x 4  
x 5  
x 6  
x 7  
x 8

x 2  
x 3  
x 4  
x 5  
x 6  
x 7  
x 8

→ import java.util.\*  
 → public class Main

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

→ Scanner scn = new Scanner (System.in);

→ int times = scn.nextInt();

→ for (int i=1; i<=times; i++)

{ int n = scn.nextInt();

// for (int i=1; i<=t; i++)  
 // { //body  
 // }

→ for statement  
 loop;

→ loop part

→ initialization

initialization,  
 increment/decrement  
 checking/condition

if →  
 comment  
 #

boolean flag = true;  
 int div = 2;  
 while (div \* div <= x)  
 { int vr = x % div;  
 if (vr == 0)

To take the input  
 from user that  
 have many times  
 the user want  
 to give a number  
 & check if its  
 prime or not prime

{ flag = false;  
 break;

}

div++;

}

if (flag == true)

{ System.out.println ("prime"); }

else { System.out.println ("not prime"); }

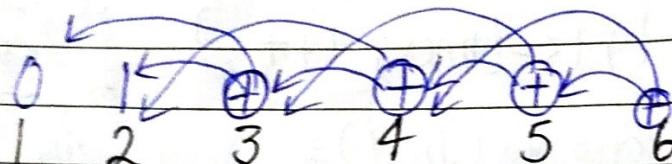
}

HW → Print All Prime Till N

Ques. 2 Print Fibonacci Numbers till n.

1st fibonacci no. → 0  
2nd " " → 1

1st fibonacci digit ०  
० दोषी ३/२ 2nd  
fibonacci digit १  
१ दोषी ।



0	1	1	2	3	5	8	13	21
1	2	3	4	5	6	7	8	9

$$(x-1)\text{th fib} + (x-2)\text{th fib} = x\text{th fib.}$$

$n$ th fibonacci number  $\equiv$   $(n-1)$ th fibonacci no. +  $(n-2)$ th fibonacci no. का sum  $\equiv$   $a+b$   
int a=0;  
int b=1;

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

$$\text{int } c = a + b;$$

System.out.println(a);

$$a=b;$$

$$b=c;$$

}

## Burit fibonacci till No java

PAGE NO.

DATE

```
→ import java.util.*;
```

```
→ public class Main
```

```
{
```

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

```
        Scanner scn = new Scanner (System.in);
```

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

```
        int a = 0;
```

```
        int b = 1;
```

```
for (int i = 1; i <= n; i++)
```

```
{
```

```
    int c = a + b;
```

```
    System.out.print (a + " ");
```

```
    a = b;
```

```
    b = c;
```

```
}
```

```
}
```

```
}
```

HW Ques

Ques \* Print all prime till n

There is a point  
no. n prime  
all primes from 2 to n.

```
→ import java.util.*;
→ public class Main
{
    public static void main (String [] args)
    {
        Scanner scan = new Scanner (System.in);
        int n = scan.nextInt();
        for (int i = 2; i <= n; i++)
        {
            int digit = i;
            boolean flag = true;
            for (int j = 2; j * j <= i; j++)
            {
                int divisor = j;
                if (i % divisor == 0)
                {
                    flag = false;
                    break;
                }
            }
            if (flag == true)
            {
                System.out.print (i + " ");
            }
        }
    }
}
```

Dec-4

# DSA Lecture = 4

PAGE NO.

DATE

Lec 3 HW → Print All Prime till n. → Solution by Sieve

Print / Print all prime within a given range  
Take a Input 'Low' as the Lower Limit of Range  
Take a Input 'High' as the higher Limit of Range  
low & high तक तक Loop (C/C++)  
जोके वीरु तक जिनके primes तक, तक  
वहाँ से print कर दो।

→ import java.util.\*;

→ public class Main

{  
public static void main (String [] args)

{  
Scanner scan = new Scanner (System.in);

int low = scan.nextInt();

int high = scan.nextInt();

for (int n = low ; n <= high ; n++)

{  
boolean isPrime = true;

int div = 2;

while (div \* div <= n)

{  
int rem = n % div;

if (rem == 0)

{  
isPrime = false;

break;

div++;

}

if (isPrime == true)

{  
System.out.println (n); } }

Count Digit = 65784383.

परे एक number दिया गया है, हमें बताना है कि इसकी digit होंगी number में।

Constraints

$$1 < n < 10^{19}$$

जब भी हमें ये constraint पैरव जाए नहीं तो इसका मतलब होगा कि हमें int use करना है।

$$\boxed{\text{Number} = 453827}$$

1) → Divide by 10 → Quotient मिलजाएगा। मतलब last की digit हो जाएगी।

2) → Count इसलो 0, और जैसे ही इसका divide किया, 1 digit का count कर दिया। Count को बढ़ावा 1 से → ऐसे ही Count को तबतक बढ़ाना होगा जब तक Number 0 ना होजाए।

10	4 5 3 8 2 7	count
10	4 5 3 8 2 → 7	1
10	4 5 3 8 → 2	2
10	4 5 3 → 8	3
10	4 → 3	4
10	4 → 5	5
10	0 → 4.	6.

→ import java.util.\*;  
→ public class Main{

PAGE NO. \_\_\_\_\_  
DATE \_\_\_\_\_

public static void main (String [] args)

{  
→ Scanner scn = new Scanner (System.in);  
→ int n = scn.nextInt();  
→ int digit = 0;  
→ while (n != 0)

{  
→ int quo = n / 10; // / (divide) → sign quotient देता है।

→ int rem = n % 10; % (Modulo) → sign remainder देता है।

n = quo;

dig++;

}  
→ System.out.println (dig);

2483 On Dividing by 10

number      quotient      remainder      count

2483            248            3            0

248            24            8            1

24            2            4            2

0            0            2            3

Q.  
J.

→ Sample Input

65784383

→ Sample Output

6

5

7

8

4

3

8

3

ये एक number है।

इस number में कुल digits हैं।

इस number की कुल digits

को as it is order में अलग।

अलग लेने और print करो।

वह digit Left to right वाली है।

number में,

उसे Top to Down अलग। अलग

लेने में print करने का काम

हमें करना है।

2 3 4 7

Divide by 10

2 3 4 | 7  
quotient      remainder

ये एक number है।

इसको 10 से divide करोगे

तो 7 अलग हो जाएगा अतः इसे

Last digit अलग होगा।

हमें ये नहीं करना X.

\* हमें तो 2 अलग करना है।

✓ 2 | 3 4 7

यो तो 2 quotient याहिं और 347 as a  
remainder तो ये पाने के लिए हमें 10 से  
पहली 1000 से divide करना होगा।

$n \rightarrow 23475$

तो number है 23475 तो इसका divide करेगी 10000 से

10000	2 3 4 7 5	0
1000	(2) - 3 4 7 5	2
100	(3) - 4 7 5	3
10	(4) - 7 5	4
1	(7) - 5	7
	5 - 0	5

quotient को print  
करवो और remainder  
को बनावो तो number

$$2 | 3475 \quad [10000].$$

$$3 | 475 \quad [1000].$$

$$4 | 75 \quad [100]$$

$$7 | 5 \quad [10]$$

5. [1]

पहली digit का no. है, जोकी digit  
का divisor आवा चाहिए।

How to get 10000 as the first divisor?

→ Divisor की शुरूआत होती है 10000 से और

पर आगे एक 0 कम होती जा रही है।

तो शुरू वाला 10000 कैसे लाया / कैसे पढ़ा चलेगा

अब शुरू में किसी divisor करना है।

div=1

जब तक n 10 या  
उससे कम जाही होता

वर तक div को

10 से multiply

करते ही जितना

वर 1000 से

जाएगो तो

10 या उससे कम

बनाने के लिए।

```
int temp=num;
int div=1;
while(temp>10)
{
    temp=temp/10;
    div*=10;
}
```

10	2 3 4 7 5
10	2 3 4 7 - 5
10	2 3 4 - 7
10	(2) - 4
	2 - 3

$\begin{matrix} \swarrow \\ \times 10 \end{matrix}$   
 $\begin{matrix} \swarrow \\ \times 10 \end{matrix}$

जोकी दी नो. less than

या equal to 10

में 10 के बीच है तो  
इसका नहीं होता  
रख जाओ।

धृष्टान्, उसके पास div को 10 से multiply करना बहुत कठोर

while ( $due \geq 1$ )

{

int quo = num / due;

int rem = num % due;

System.out.println (quo);

num = rem;

due = due / 10;

div	1000	100	10	1	0	num
	1	2	3	4	7	
		1	2	3	4	7
			3	4	7	num
			1	4	7	num
				7	-	0 num

→ import java.util.\*;

→ public class Main {

→ public static void main (String [args])

{

→ Scanner scan = new Scanner (System.in);

→ int n = scan.nextInt(); Dry run for 2347

→ int div = 1;

→ int temp = n;

→ while (temp >= 10)

{

temp = temp / 10;

div \*= 10;

10	2	3	4	7	1
10	2	3	4		10
10	2	3			100
2					1000

→ while (due >= 1)

{

int q = n / due;

int r = n % due;

System.out.println (q);

n = r;

due = due / 10;

}

}

1000	2	3	4	7	n
100	2	3	4	7	n
10	3	4	7	n	
1	4	7	n		
0	7	-	0	n	

CW3

## INVERSE OF A NUMBER

PAGE NO.

DATE

Ques

1. You are given a no. with following constraints:-

i) → The key constraint is if the number is 5 digits long, it will contain all the digits from 1 to 5 without missing any and without repeating any. e.g. → 23415 is a 5 digit no. containing all the digits from 1 to 5.

\* Few valid numbers are :- a) 624135 (bcz its a 6 digit no. & 1-6, all the digits are present)

b) 81456273 → Its a 8 digit no., & 1, 2, 3, 4, 5, 6, 7, 8 → all of these digits are present here.

\* Few Invalid numbers are :- 139, 7421357, etc.

→ The inverse of a number is defined as the no. created by interchanging the index & the face value of a no.

e.g. → 426135 (reading from right to left,  
5 is in place 1.  
3 is in place 2  
1 is in place 3  
6 is in place 4  
2 is in place 5.  
4 is in place 6 )

The Inverse will be 416233 (reading from right to left  
3 is in place 1  
5 is in place 2  
2 is in place 3  
6 is in place 4  
1 is in place 5  
4 is in place 6 )

eg 2) → Inverse of 2134 is 1243

eg 3) → Inverse of 24153 is 24153.

2. → Now take a number "n", assume that the numbers will follow these constraints

3. → Print its inverse

## Solution

$$\begin{array}{c} \boxed{4 \ 2 \ 6 \ 1 \ 3 \ 5} \rightarrow \boxed{4 \ 1 \ 6 \ 2 \ 5 \ 3} \\ \text{Index} \rightarrow 6 \ 5 \ 4 \ 3 \ 2 \ 1 \end{array}$$

- Original number में अंक 1st Pos <sup>n</sup> पर 5 है।  
अंक वाले से जो उत्तर में 5th Pos <sup>n</sup> पर 1 है। आउटपुट। -
- Original number में अंक 2nd Pos <sup>n</sup> पर 3 है।  
अंक वाले से जो उत्तर में 3rd Pos <sup>n</sup> पर 2 आउटपुट। -
- Original number में 3rd Pos <sup>n</sup> पर 1 है।  
अंक वाले से जो उत्तर में 1st Pos <sup>n</sup> पर 3 आउटपुट। -
- Original number में 4th Pos <sup>n</sup> पर 6 है।  
अंक वाले से जो उत्तर में 6th Pos <sup>n</sup> पर 4 आउटपुट। -
- Original number में 5th Pos <sup>n</sup> पर 2 है।  
अंक वाले से जो उत्तर में 2nd Pos <sup>n</sup> पर 5 आउटपुट। -
- Original number में 6th Pos <sup>n</sup> पर 4 है।  
अंक वाले से जो उत्तर में 4th Pos <sup>n</sup> पर 6 आउटपुट। -

$$\begin{array}{c} \boxed{2 \ 4 \ 1 \ 3} \rightarrow \boxed{3 \ 1 \ 4 \ 2} \\ \text{Index} \rightarrow 4 \ 3 \ 2 \ 1 \end{array}$$

सबसे पहले Code लिखते ही time को check करना होगा क्या वो no. valid एवं Inverse गिनतीने के लिए?  
n digit no. must have all the digits (1 to n) in it

$$\begin{array}{r}
 5 \ 1 \ 4 \ 2 \ 3 \\
 5 \ 4 \ 3 \ 2 \ 1 \\
 \downarrow \\
 \end{array}$$

$$\begin{array}{r}
 5 \ 3 \ 1 \ 2 \ 4 \\
 5 \ 4 \ 3 \ 2 \ 1 \\
 \end{array}$$

$$\begin{array}{r}
 5 \ 1 \ 3 \ 2 \ 6 \ 4 \\
 6 \ 5 \ 4 \ 3 \ 2 \ 1 \\
 \downarrow \\
 \end{array}$$

$$\begin{array}{r}
 2 \ 6 \ 1 \ 4 \ 3 \ 5 \\
 6 \ 5 \ 4 \ 3 \ 2 \ 1 \\
 \end{array}$$

$$893 = 8 \times 10^2 + 9 \times 10^1 + 3 \times 10^0$$

100      10      1

$$\begin{array}{r}
 3 \ 5 \ 2 \ 1 \ 4 \\
 5 \ 4 \ 3 \ 2 \ 1 \\
 \end{array}$$

$$\begin{array}{r}
 3 \ 5 \ 2 \ 1 \ 4 \\
 \times \quad \times \quad \times \quad \times \quad \times \\
 10^4 \quad 10^3 \quad 10^2 \quad 10^1 \quad 10^0 \\
 \end{array}$$

$$\begin{array}{r}
 4 \ 1 \ 5 \ 3 \ 2 \\
 5 \ 4 \ 3 \ 2 \ 1 \\
 \end{array}$$

$$\begin{array}{r}
 4 \ 1 \ 5 \ 3 \ 2 \\
 \times \quad \times \quad \times \quad \times \quad \times \\
 10^4 \quad 10^3 \quad 10^2 \quad 10^1 \quad 10^0 \\
 \end{array}$$

$$\begin{array}{r}
 4 \ 1 \ 5 \ 3 \ 2 \\
 \times \quad \times \quad \times \quad \times \quad \times \\
 10^4 \quad 10^3 \quad 10^2 \quad 10^1 \quad 10^0 \\
 \hline
 4 \ 1 \ 5 \ 3 \ 2 \\
 1 \ 2 \ 3 \ 4 \ 5 \\
 \hline
 4 \ 1 \ 5 \ 3 \ 2
 \end{array}$$

मतलब  
 4 का 10<sup>4</sup> पर  
 1 का 10<sup>3</sup> पर  
 2 का 10<sup>2</sup> पर  
 3 का 10<sup>1</sup> पर  
 5 का 10<sup>0</sup> पर

$$\begin{array}{r}
 1 \times 10^3 = 1000 \\
 2 \times 10^0 = 2 \\
 3 \times 10^1 = 30 \\
 4 \times 10^4 = 40000 \\
 5 \times 10^2 = 500 \\
 \hline
 + 41532
 \end{array}$$

इसका मतलब  
 4 का 10<sup>4</sup> पर  
 1 का 10<sup>3</sup> पर  
 2 का 10<sup>2</sup> पर  
 3 का 10<sup>1</sup> पर  
 5 का 10<sup>0</sup> पर

तब उन सभी को multiply करके results को add करते पाओ, last sum करते ही  
 हम answer लिख सकते हैं।

Scanner scn = new Scanner (System.in);

int n = scn.nextInt();

Answer → int univ = 0;

int p = 1;

while ( $n > 0$ )

{ int rc = n % 10;

int q = n / 10;

n = number

p = place

rc = remainder

q = quotient

univ = univ +

↓  
answer.

$$n = q \cdot p + r$$

// हमें  $r$  मिला हैः pth pos<sup>n</sup> QR,

हमें  $p$  मिला हैः 10<sup>r</sup> with pos<sup>n</sup> QR,

तो  $p * \text{pow}(10, r-1)$  करने से होगा।

//  $r$  <sup>मिला QR</sup>  $\rightarrow$   $p$  <sup>मिला QR</sup>  $\rightarrow$   $p * \text{pow}$  <sup>Math</sup>  $(10, r-1)$

univ = univ + p \* (int) Math.pow(10, r-1);

↑ जो हमने

Explicit Type

Conversion की जरूरी Math.pow  
result float हो देता है

$p++;$   
y

System.out.println(univ);

g g

```
→ import java.util.*;  
→ public class Main {  
→     public static void main (String [] args) {  
→         Scanner scan = new Scanner (System.in);  
→         int n = scan.nextInt();  
→         int p = 1;  
→         int univ = 0;  
→         while (n > 0) {  
→             int r = n % 10;  
→             n = n / 10;  
→             univ = univ + p * (int) Math.pow (10, r - 1);  
→             p++;  
→         }  
→     }  
→ }
```

$n = q \cdot 10^r$   
//  $q @ p \rightarrow p @ r \rightarrow p * \text{pow}(10, r-1)$   
उम्मीद पर प्राप्ति रप्त करना होगा

univ = univ + p \* (int) Math.pow (10, r - 1);

$p++;$

System.out.println (univ);

HW Ques

## Reverse a Number

PAGE NO.

DATE

Sample → 6 5 7 8 4 3 8 3

Input

Sample Output → 3 (last digit at top)

8

3

4

8

7

5

6 (first digit at bottom)

1 = gives quotient

% = gives remainder

एक loop लगाते हैं तबतक चलेगा तब तक num 0 नहीं हो जाता है। हर intery में loop में, remainder किसी भी point करते जाएगा और quotient को number घनाते जाएंगे, थे तबतक continue होता तब तक num 0 नहीं हो पाता

$$\begin{array}{r}
 n = 65784383 / 10, \text{ quotient } 6578438, \text{ remainder } 3 \\
 n = 65784383 / 10, \text{ quotient } 657843, \text{ remainder } 8 \\
 n = 6578438 / 10, \text{ quotient } 657843, \text{ remainder } 8 \\
 n = 657843 / 10, \text{ quotient } 65784, \text{ remainder } 3 \\
 n = 65784 / 10, \text{ quotient } 6578, \text{ remainder } 8 \\
 n = 6578 / 10, \text{ quotient } 657, \text{ remainder } 8 \\
 n = 657 / 10, \text{ quotient } 65, \text{ remainder } 7 \\
 n = 65 / 10, \text{ quotient } 6, \text{ remainder } 5 \\
 n = 6 / 10, \text{ quotient } 1, \text{ remainder } 6 \\
 n = 1 / 10, \text{ quotient } 0, \text{ remainder } 0
 \end{array}$$

→ import java.util.\*;  
→ public class Main  
    {  
        → public static void main (String [] args)  
            {  
                → Scanner scan = new Scanner (System.in);  
                → int n = scan.nextInt();  
                → while (n != 0) → जब तक n 0 नहीं होता  
                    { int q = n / 10;      टेंडर  
                    { int r = n % 10;      रुपये  
                → System.out.println (r);  
                → n = q;  
            }  
    }

HW Ques 2

## Rotate a number

number = 2 3 4 5 7

1) rotate करना।  
इसे तो last के 1 rotation + 1 = 7 2 3 4 5

digit को आगे लाना है।

rotation + 2 = 5 7 2 3 4

2) rotate

करना है मतलब last के

2 digit को आगे लाना है।

3) rotate करना है।

मतलब last rotation + 3 =

की 3 digit को आगे लाना है।

4 5 7 2 3

& so on } 1 से rotate करना है मतलब 10 से divide कर  
2 से rotate करना है मतलब 100 से divide कर  
3, " " " " " " " " 1000 " " " " " "

2 से rotate करना। 97543 को तो 100 से divide कर

$K = 2^{(10^2)}$ , Divisor

$n - K = 3^{(10^3)}$ , Multiplier

no. of digit in number

9 7 5 | 4 3

43975 → it answer

पाइये

quotient = 975

remainder = 43

अब इसका 43975 बनाने के लिए

43 को 1000 से multiply करो और  
उसमें 975 add करदो।

$$43 \times 1000 + 975 = 43975$$

number → the given number.

$n$  = no. of digit

$K$  = Rotate by  $k$

$q$  = quotient

$r$  = remainder

Rotate by 2 =  ~~$2^{\text{nd}}$  Difierior~~  
 no. of digits in  $n = 6$ .

9754 | 32

$$\text{Divisor} = 100 = (10^k)$$

$$n-k=6-2=4$$

10^4 = Multiple

$$\begin{aligned}
 & (104 = \text{Multiples}) \\
 & = (\text{Remainder} * \text{Multiples} + \text{Quotient}) = \text{Answer} \\
 & = (32 * 10000) + 9754 \\
 & = 329754.
 \end{aligned}$$

```

→ import java.util.*;
→ public class Main {
→   public static void main (String[] args) {
    Scanner scan = new Scanner (System.in);
    int num = scan.nextInt();
    int k = scan.nextInt(); // num = number
    int nod = 0; // K = rotate by K
    int temp = num; // nod = no. of digits in num
    while (temp != 0) {
        if (temp == 0) break;
        temp = temp / 10; // import part
        nod++; // side cases
    }
    int div = 1; // K = K % nod
    if (k == 0) k = k + nod; // equal 25
    int mult = 1; // if k = 9, K = 9 + 5 = 4
    for (int i = 1; i <= nod; i++) {
        if (i <= k) {
            div = div * 10; // num = 67532, rotation
        } else {
            mult = mult * 10;
        }
        int q = n / div; // 6
        int r = n % div; // 7
        int ans = (r * mult) + q; // 8
        System.out.println (ans); // 9
    }
}

```

if  $K = 25$   
 $K = 25 \% 5 = 0$   
 $\text{if } k = 9, K = 9 + 5 = 4$

digit      2    1    =    7  
             3    1           8  
             4    1           9

Negative rotation  
 अंकों की विपरीत संग्रहीत  
 प्राप्त उत्तर (Ans)

Negative Rotations में  $\text{अंकों}$  की digits पीछे चली जाती हैं।

-5	5	6	7	4	3
-4	3	5	6	7	4
-3	7	3	5	6	7
-2	7	4	3	5	6
-1	6	7	4	3	5
0	5	6	7	4	3
+1	3	5	0	7	4
+2	4	3	5	6	7
+3	7	4	3	5	6
+4	6	7	4	3	5
+5	5	6	7	4	3

आगे से 1 digit पीछे आ जाएँ  $\Rightarrow$  पीछे से 4 digit आगे आ रहे हैं।