

Lecture-1How to print in Java

`System.out.print ("Hello") ;` → To print  
 → इस line Java में  
 Semi colon पर  
 end एवं है।

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

A) Skill language → Output

of an lang System.out.println ("Hello");

↳ Print line = println

1) `import java.util.*`

`public class Main {` ↳ Recoding IDE में class का नाम  
 Main हो जाएगा where M is capital

`public static void main (String [] args)` ↳ write inside main initially

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

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

3

Output →

Hello world Hello world

• यही output 2 alag alag line के रूप से था  
 लेकिन मिला सकता है।

\* To print next message in next line

|      |  |  |
|------|--|--|
| DATE |  |  |
|------|--|--|

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

3)

Next line में print करने के लिए इस रिकॉर्ड print का भी use कर सकते हैं।

- 1st line System.out.print ("Hello world \n");  
↓  
Back slash n,  
इससे नेक्स्ट लाइन का special character जाता है या नेक्स्ट लाइन में enter लग जाता है।  
2nd line System.out.println ("Hello world");

→ Output → HelloWorld  
Hello world.  
This anyways is not going to make any difference 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 (" * * * * * ");

```

Output

अगर दो रेंट दी print use करते हैं तो

```

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

```

→ \n does its work  
of pressing  
an enter  
key

Way 2

```

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

```

Ques. 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.

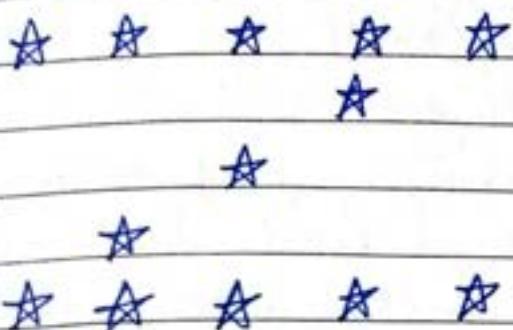
public static void main()

Key 1  
using System.out.println (" \* \* \* \* \* ");  
System.out.println (" \_ \_ \_ \* ");  
System.out.println (" \_ \_ \* ");  
System.out.println (" \_ \* ");  
System.out.println (" \* \* \* \* ");  
Space

Ar.

ways 2  
using System.out.println (" \*\*\* \* \*\n \* \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

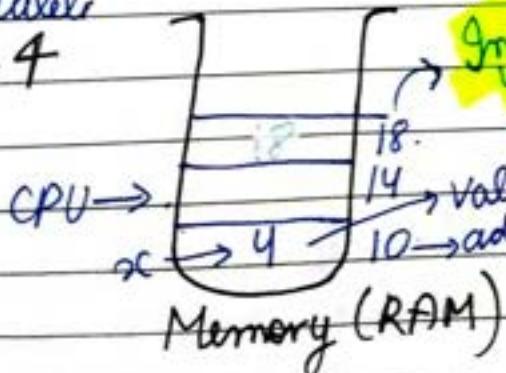
5 Skills of a Language

## 2nd Skill → Variables

Hard Disk keeps the data permanent even if Computer

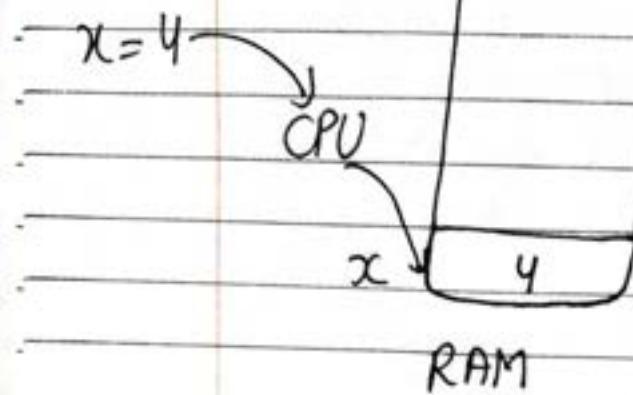
Variables

$$x = 4$$



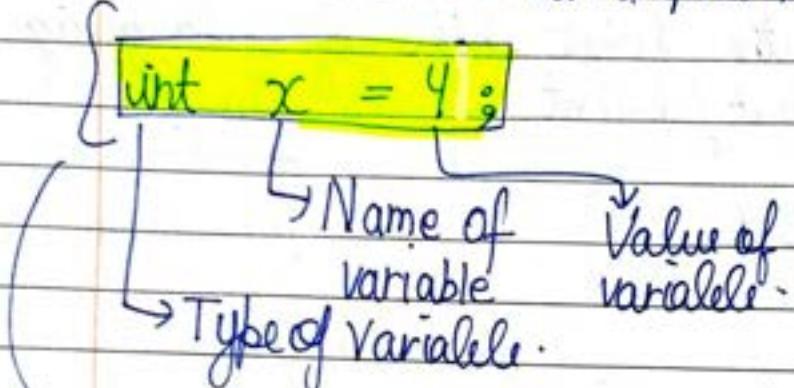
In Java, if we need to worry about address, HD will stay.

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



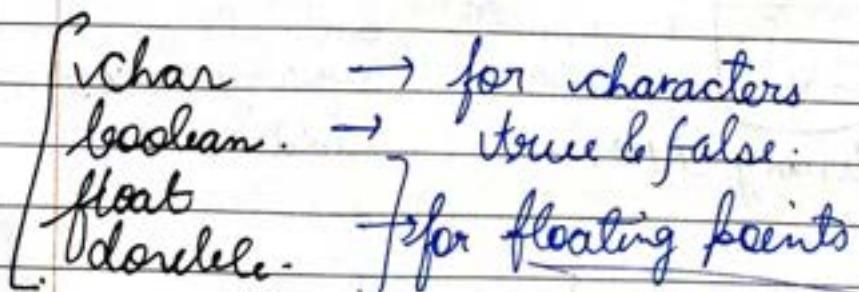
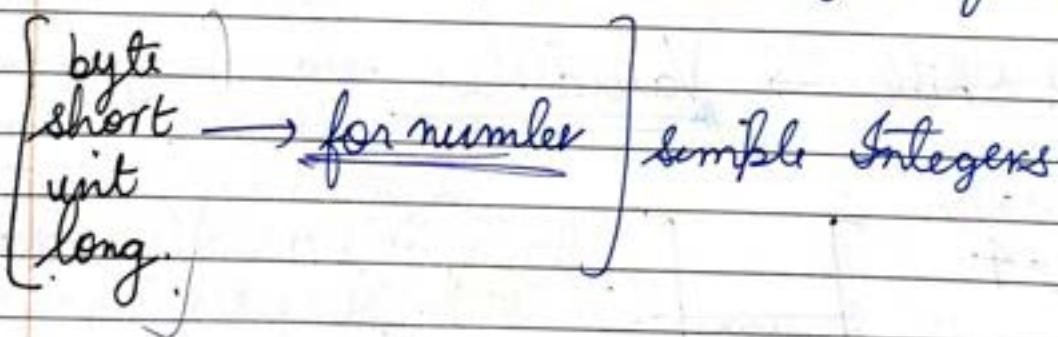
CPU has one eye on program & another on RAM.

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



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

`int` → Most popular datatype of number

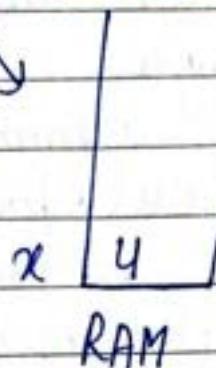


CPV / System / Computer

keep one eye  
on program

and another  
eye on RAM

```
int x=4;  
System.out.println(x);  
System.out.println("x");
```



It will

print x  
verbatim

Verbatim

means as  
it is.

Console

Output → 4.

CW.

1. How to Print in Java :

Skill 1.

Ques.

2. How to print Z :

3. " " " Pattern of Star :

Skill 2

4. What are Variables & How to print

HW

In Getting Started Module

Video

1. How to Print in Java

Article

2. How to print in Java.

Ques.

3. Print Z.

Video

4. Variables

Article

5. Variables in Java

MCQ.

6. Variables in Java

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 कौना पता है कि उस data का datatype है।

उदाहरण करने की बात अच्छी, लेकिन तो हम datatype की जांच करेंगे।

int x = 15;

int y = 10;

int sum = x + y;

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

y  
x

10.

15

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

Sum

x +

" and " +

y +

" is " +

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

Divide

&

Quotient

System.out.println(v1);

System.out.println(v2);

System.out.println(v3);

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

PAGE NO.

DATE

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

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

equal priority      equal priority

$\text{int } x = 10 ;$   
 $\text{int } y = 15 ;$

If equal then  
start from left to  
right.

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

Without brackets

$$= 10 * 15 / 15 + 10 .$$

$$= 150 / 15 + 10 .$$

$$10 + 10 = 20 \quad \checkmark$$

With brackets

$$= (10 * 15) / (15 + 10)$$

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

## 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  
guru

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);
```

→  $a=10 \quad b \neq 20$

$a=30 \quad b=10$

$a=20$ .

$a=20, b=10;$

|   |    |    |    |
|---|----|----|----|
| a | 20 | 20 | 20 |
| b | 10 | 10 | 10 |

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>

3.

✓ — 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 when 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

2.5. " " " " than equal to 60, expectation

```
if (marks > 90)
```

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

```
} else
```

```
{ if (marks > 80 && marks <= 90)
```

→ and = 37%  
else

```
{ 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"); }
```

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

```
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  $16 \% 3 \rightarrow 1$ .

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

$$\begin{array}{r} 1 \\ \hline 2 \sqrt{21} \\ 20 \\ \hline 1 \end{array} \rightarrow \text{remainder } \neq 0 \\ \therefore \text{Odd.}$$

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

$$\begin{array}{r} 0 \\ \hline 2 \sqrt{20} \\ 20 \\ \hline 0 \end{array} \rightarrow \text{remainder } = 0. \\ \therefore \text{Even}$$

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

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

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

```
}
```

```
close
```

```
{ 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. 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"); }
```

```
else {
```

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

{ 3K

}

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

{ 3K+1

}

else

{ 3K+2 .

}

Commented Code never runs

Single line comment  $\rightarrow //$

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

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");

}

## 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

Print 1 to 5

Console

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

1  
2  
3  
4  
5  
done.

RAM

System.out.println("done");

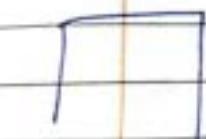
## 5th) Skill Input

PAGE NO.

DATE

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

Scanner scn = new Scanner (System.in); To take input  
Variable of type Scanner.  
Scanner is a used to read  
from keyboard



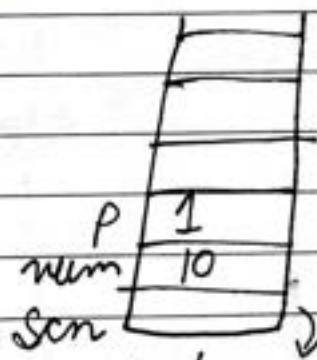
```
Scanner scn = new Scanner (System.in);
int num = scn.nextInt();
System.out.println (" You entered " + num);
```

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<br>2<br>3<br>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 printoddoreven {
    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-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.\*; → ReCodding editor में ये  
line कैसे delete होना चाहिए और class name  
हमें Main ही लिखेगा।

public class Main {

public static void main  
(String[] args)

↑ Capital M

→ codechef QR शीर्षीयों द्वारा दी गई है।

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

पहुँचते ही CPU

Scanner  
type of object  
variable को समझते  
हैं जो input  
read करते  
ram में assign  
करते हैं।

int p = 1;

while (p <= n)

terminal / console को दर्शगा

अब वहाँ से num पढ़ा दूँ।

int rem = p % 2;

अब वहाँ से उताके न नाम के

if (rem == 0)

variables में डाल दिया

जाएगा और ram में store

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

कर दिया जाएगा।

else

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

| Input | Output    | Scn     |
|-------|-----------|---------|
| 7     | 1 is odd  | 7       |
|       | 2 is even | P       |
|       | 3 is odd  | num     |
|       |           | Scanner |

## Q.1 Is a number prime

### Way 1

तो कहि भी no. prime तब होता है जो वो 1 और खुद 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  
∴ 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 divisible हो पा रहा है नहीं

→ So Prime no. बस वही होते हैं जो 1 से n तक केवल 2 एवं इनके गुणों से नहीं। → So Way 1 है check करने के लिए

→ अगर 2 - (n-1) तक इस बार भी divisible ना हो तो no. होगा prime.

Checking 11 by this Way 1.

|     |     |      |
|-----|-----|------|
| 2 × | 5 × | 8 ×  |
| 3 × | 6 × | 9 ×  |
| 4 × | 7 × | 10 × |

2 से (n-1) तक कहि भी divisible नहीं कर पाया।

∴ 11 is Prime

**Way 3.**

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

→ Check 50 कोन-कोनसे no. से divide होता है।

50

1 ✓

2 ✓

5 ✓

10 ✓

25 ✓

→ Check 6 कोन-कोनसे no. से divide होता है?

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

6

1 ✓

2 ✓

3 ✓

तो 6 के बाद किसी भी 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 दी factor मिलसकता है तो n का  
उसके बाद तो n दी होता है।

→ Even no. can never be prime except 2

/ → gives quotient

% → gives remainder

→ तो हमें वह  $\frac{x}{2}$  से  $\frac{x}{2}$  तक दी check करने की जरूरत

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

## Way 4

हमें 2 तक  $\left(\frac{x}{2}\right)$  तक check करने की जरूरत नहीं है।

\* Check factorization of 36.

Unique factors

$$1 \times 36 = 36$$

$$36 \times 1 = 36$$

$$2 \times 18 = 36$$

$$18 \times 2 = 36$$

$$3 \times 12 = 36$$

$$12 \times 3 = 36$$

$$4 \times 9 = 36$$

$$9 \times 4 = 36$$

Mirror  
Image

$$6 \times 6 = 36$$

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

\* Check factorization of 25.

$$1 \times 25$$

$$5 \times 5$$

$$25 \times 1$$

Square root के around mirror image

\* Check factorization of 64.

जो पहला Quotient हो, mirror image of divisor होगा।

$$1 \times 64$$

$$64 \times 1$$

$$2 \times 32$$

$$32 \times 2$$

$$4 \times 16$$

$$16 \times 4$$

$$8 \times 8$$

Square root के around mirror image हैं।

p, q का Product  $\leq 64$  हो, p & q both cannot be greater than 7.

\* Check factorization of 35

$$\sqrt{35} = 5 \times$$

something

$\frac{2}{3} \times \frac{3}{4} \times \frac{4}{5} \times \frac{5}{\checkmark}$

→ तो 5 तक ही इह check करें। दोगला divide मतलब non prime  
और  $p, q$  दोनों  $\sqrt{n}$  से बड़े नहीं  
हो सकते। या तो  $p$  और  
 $q, \sqrt{n}$  के equal हो सकते  
या  $p$  वृत्ति  $q$  दोषा  $\sqrt{n}$  से  
 $q < \sqrt{n}$ . या vice versa.

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

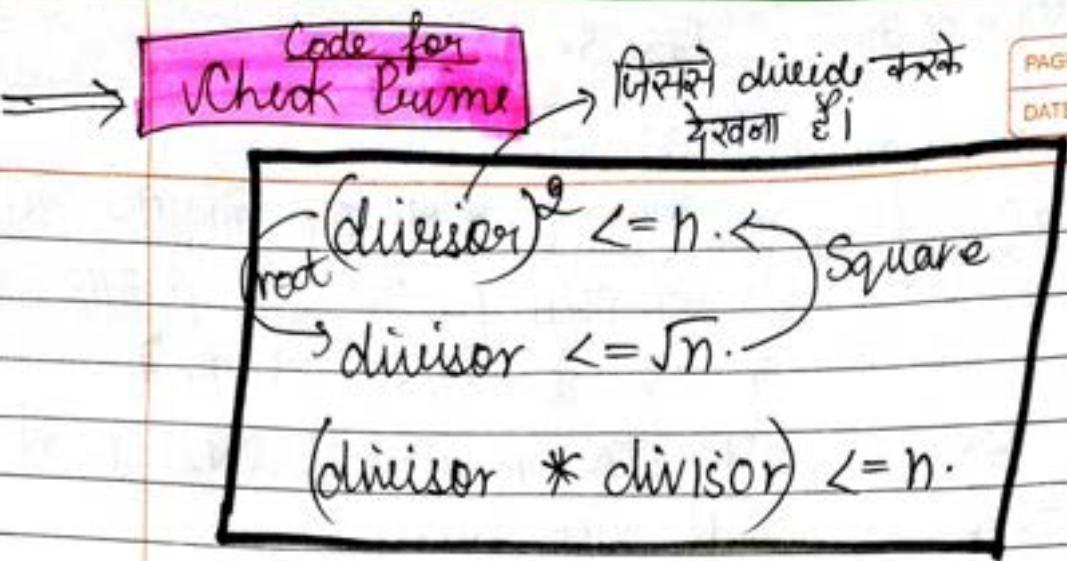
\* Prime check करने के ways

→ Way 1 → [1 to  $x$ ] → 1 to  $x$  तक divide करें।  
और वस्तु 2 से divide होता है  
मतलब 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 को  $\frac{x}{2}$  के बाद कोई  
factor नहीं लिया जाए except  $x$ .  
जो अगर 2 से  $\frac{x}{2}$  तक कोई factor  
न लिये तो  $\frac{x}{2}$  मतलब  $x$   
prime है।

→ Way 4 (2 to  $\sqrt{x}$ ) → 2 से  $\sqrt{x}$  तक ही check करें।  
यदों  $\sqrt{x}$  के बाद कोई नया no. divide  
नहीं करता, और पहले 36 को 4 ने divide  
किया है उसके 9 लगाके तो बादमें  $\sqrt{36}$  के  
9 divisor चलेगा, 4 quotient भी कोई  
नया no. नहीं लियेगा। 2 से  $\sqrt{x}$  तक किसी से  
divide नहीं हुआ न तो वो prime है।



→ int div = 2 ;  
 → boolean flag = true ; // assume कि यह prime है।  
 → while (div \* div <= n)

{ int rem = n % div ;

if (rem == 0)

{ flag = false ;

break ; → break loop को तुरंत छीच में दी जाती करेगी है।

div ++ ;

}

35

→ if (!flag == true)

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

else

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

or 35

x 35

div 2 3 5

flag true/false

35 is not prime

x=53

flag

true

2

3

4

5

6

1

2

3

4

5

6

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

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

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

→ unit stories = `scr.nextInt();`

→ `for( int i=1; i<=times; i++ )`

{ int n = Sch.nextInt(); }

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

```

boolean flag = true ;
int dw = 2;
while (dw * dw <= x)
    if (x - dw * dw == 0)
        flag = false;

```

```
{ flag = false;  
    break;  
}  
list++;
```

```
if(flag == true){  
    System.out.println("prime");  
} else {  
    System.out.println("not prime");  
}
```

7. L-systems

962

*fff*

*Journal of Health Politics, Policy and Law*, Vol. 27, No. 4, December 2002  
ISSN 0361-6878 • 10.1215/03616878-27-4 © 2002 by The University of Chicago

—  
—

Digitized by srujanika@gmail.com

→ far ~~मिल~~  
लोक्य

एक loop में

## ચોણ હોલો

initialization,  
increment/decrement  
checking/condition

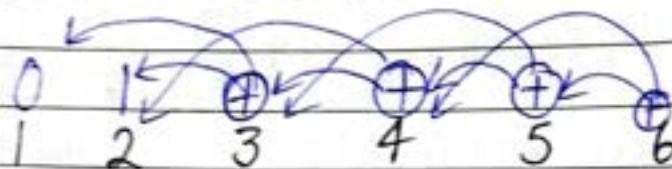
To take the input from user that how many times the user want to give a number & check if it's prime or not using

HW → Print All Prime Till N

Ques. 2 Print Fibonacci Numbers till n.

1st fibonacci no. → 0  
2nd "      n. → 1

1st fibonacci digit ० १  
० और १ की 2nd  
fibonacci digit ३ ५  
१ और १ की



|   |   |   |   |   |   |   |    |    |
|---|---|---|---|---|---|---|----|----|
| 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<sup>th</sup> fibonacci number  $\infty$  (n-1)<sup>th</sup> fibonacci no.  
 $\infty$  (n-2)<sup>th</sup> fibonacci no.  $\infty$  sum  $\infty$   
 init a=0;  
 init b=1;

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

```
{  
    int c = a+b;  
    System.out.println(a);  
    a=b;  
    b=c;  
}
```

}

## Burit fibonacci till N 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 print 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 = 1;  
            boolean flag = true;  
            for (int j = 2; j * j <= i; j++)  
            {  
                if (i % divisor == 0)  
                {  
                    flag = false;  
                    break;  
                }  
            }  
            if (flag == true)  
            {  
                System.out.print(i + " ");  
            }  
        }  
    }  
}
```

# DSA Lecture = 4

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

Put / Print all prime within a given range  
 Take Input 'Low' as the lower limit of Range  
 Take Input 'High' as the higher limit of Range  
 low & high तक तक Loop (लूप) करें,  
 जोके बीच में उनमें से primes आएँ, उन्हें print करें।

→ 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

पैर एक number दिया गया है, हमें बताना है कि इसकी digit हैं इस number में।

Constraints

$$1 \leq n \leq 10^{19}$$

जब भी हमें ये constraint प्रिय जाए ना तो इसका मतलब होगा कि हमें int पर करना है।

Number = 453827

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

2) → Count रखतो 0 और जैसे ही सभावार 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 हो जाए।

|    |               |       |
|----|---------------|-------|
| 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 5 → 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) का sign quotient देता है।

int rem = n % 10; % (Modulo) का 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

0            0            0            4

→ Sample Input

65784383

→ Sample Output

6

5

7

8

4

3

8

3

ये स्कॉफ़ नंबर हैं।

इस नंबर में कुल ही digits हैं।

इस नंबर की कुल ही digits

को as it is order में लिखा।

अलग लाइन पर print करो।

ये digit Left to right पर रखी हैं।

नंबर में,

उन्हें Top to Down अलग अलग

लाइन में print करने का काम

हमें करना है।

2 3 4 7

ये स्कॉफ़ नंबर हैं।

Divide by 10

इसको 10 से divide करोगे।

2 3 4 | 7  
quotient      remainder

तो 7 अलग हो जाएगा भितलवे।  
Last digit अलग होगा।

हमें ये नहीं करना X.

\* हमें तो 2 अलग करना है।

✓ 2 | 3 4 7

यो तो 2 quotient याकि ठार 347 as a  
remainder तो ये बाने के लिये हमें 10 से  
पहीं 1000 से divide करना होगा।

n → 2 3 4 7 5

तो number है 23475 तो क्षमता 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     | 5 - 0       | 5 |

quotient को print  
करवो और remainder  
को बनाये तो number

$$2 \mid 3475 \quad [10000]$$

$$3 \mid 475 \quad [1000]$$

$$4 \mid 75 \quad [000]$$

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

5. (1) *प्रथम digit का no है, तो उसी digit  
का divisor जाना चाहिए।*

How to get 10000 as the first Divisor?

→ Division की शुरूआत होती है 10000 से और  
फिर आगे एक एक 0 कम होती जा रही है।  
तो शुरू वाला 10000 कैसे लाए। कैसे पता चलेगा।  
अब शुरू में किसी divisor करना है।

```
int temp = num;
int div = 1;
while (temp > 0)
{
    temp = temp / 10;
    div *= 10;
}
```

div=1

जब तक n 10 या  
उससे कम जाही होता।  
तब तक div को  
10 से multiply  
करते रहो तितना।  
वह 1000 से  
जास्ती नहीं  
(10 या उससे कम  
बनाने के लिए।)

नो 10 के बड़े नहीं होते।  
इसका इसका

जोसे ही no. less than

या equal to 10

घराए, उसके पास due को 10 से multiply करना बहुत कठोर।

while ( $due \geq 1$ )

{

    int quo = num / div;

    int rem = num % div;

    System.out.println (quo);

    num = rem;

    due = due / 10;

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

→ 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;

10 | 2 3 4 7 | 1

    int temp = n;

10 | 2 3 4 | 10

    while (temp >= 10)

10 | 2 3 | 100

{

        temp = temp / 10;

2 | 1000

        div \*= 10;

}

    while (div >= 1)

{

        int q = n / div;

1000 | 2 3 4 7^n

        int r = n % div;

100 | 2 - 3 4 7^n

        System.out.println (q);

10 | 3 - 4 7^n

        n = r;

1 | 4 - 7^n

        div = div / 10;

0 | 7 - 0^n

}

}

CW3

## INVERSE OF A NUMBER

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

Ques

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

- a) → 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.

- 2) → 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 )

e.g. 2) → Inverse of 234 is 123

e.g. 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

|         |             |   |             |
|---------|-------------|---|-------------|
| Index → | 4 2 6 1 3 5 | → | 4 1 6 2 5 3 |
|         | 6 5 4 3 2 1 |   | 6 5 4 3 2 1 |

- Original number में अंक 1st Pos<sup>n</sup> पर 5 है।  
अंत वाले ने भी उत्तर में 5 लिया। -
- Original number में 5th Pos<sup>n</sup> पर 1 है।  
उत्तर में 1 लिया। -
- Original number में 2nd Pos<sup>n</sup> पर 3 है।  
उत्तर में 3 लिया। -
- Original number में 3rd Pos<sup>n</sup> पर 2 है।  
उत्तर में 2 लिया। -
- Original number में 3rd Pos<sup>n</sup> पर 1 है।  
उत्तर में 1 लिया। -
- Original number में 1st Pos<sup>n</sup> पर 3 है।  
उत्तर में 3 लिया। -
- Original number में 4th Pos<sup>n</sup> पर 6 है।  
उत्तर में 6 लिया। -
- Original number में 5th Pos<sup>n</sup> पर 2 है।  
उत्तर में 2 लिया। -
- Original number में 2nd Pos<sup>n</sup> पर 5 है।  
उत्तर में 5 लिया। -
- Original number में 6th Pos<sup>n</sup> पर 4 है।  
उत्तर में 4 लिया। -
- Original number में 4th Pos<sup>n</sup> पर 6 है।  
उत्तर में 6 लिया। -

|         |   |         |
|---------|---|---------|
| 2 4 1 3 | → | 3 1 4 2 |
| 4 3 2 1 |   | 4 3 2 1 |

सबसे पहले Code लिखते हीमें check करवा होगा कि  
ये no. valid & Inverse मिटालने के लिए?  
n digit no. must have all the digits (1 to n) in it

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

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

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

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

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

100      10      1

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

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

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

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

$$\begin{array}{r}
 4 \cancel{\times} \ 1 \ 92 \rightarrow 1 \text{ दोना चाहिए} \ 4 \ 92 \rightarrow 1 \times 10^3 \\
 1 \cancel{\times} \ 2 \ 92 \rightarrow 2 \text{ दोना चाहिए} \ 1 \ 92 \rightarrow 2 \times 10^0 \\
 2 \cancel{\times} \ 3 \ 92 \rightarrow 3 \text{ दोना चाहिए} \ 2 \ 92 \rightarrow 3 \times 10^1 \\
 5 \cancel{\times} \ 4 \ 92 \rightarrow 4 \text{ दोना चाहिए} \ 5 \ 92 \rightarrow 4 \times 10^4 \\
 3 \cancel{\times} \ 5 \ 92 \rightarrow 5 \text{ दोना चाहिए} \ 3 \ 92 \rightarrow 5 \times 10^3
 \end{array}$$

इसका मतलब

दोना कि 5 को multiply करके  $10^{13}$  से

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

उन सभी को multiply करके result को add करते जाओ, last sum करते ही इसी answer प्राप्त होगा।

Scanner scn = new Scanner (System.in);

int n = scn.nextInt();

n = number

Answer → int univ = 0;

p = place

int p = 1;

r = remainder

while ( $n > 0$ )

q = quotient

{

int r = n % 10;

univ = univ +

int q = n / 10;

↓  
answer.

$$n = q, r$$

// हमें r मिला हैः pth pos^n QR,

ईसे p गुणवा है और nth pos^n QR,

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

//  $\text{univ} @ p \rightarrow p @ r \rightarrow p * \text{pow}(10, r-1)$ .

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

→ यह समने  
Explicit Type

Conversion की जरूरी Math.pow  
result float में देता है।

$p++;$

System.out.println(univ);

}

```
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;
            int q = n / 10;
```

$n = q$ ,  
//  $q @ p \rightarrow p @ r \rightarrow p * \text{pow}(10, r-1)$   
अपेक्षा प पर P जोगा तो r पर r करना होगा

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

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

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 जब तक num 0 नहीं हो जाता है। हर entry में loop में, remainder बिकाले point करते जाएंगे और quotient को number में बनाते जाएंगे, ये तबतक continue होगा जब तक num 0 नहीं हो जाता।

$$\begin{array}{r}
 n = 65784383 / 10. \quad \text{quotient} \quad \text{remainder} \\
 n = 65784383 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 3 \\
 n = 6578438 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 8 \\
 n = 657843 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 3 \\
 n = 65784 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 8 \\
 n = 6578 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 7 \\
 n = 657 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 5 \\
 n = 65 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 6 \\
 n = 16 \qquad \qquad \qquad \qquad \qquad \qquad \qquad 0
 \end{array}$$

```
→ 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) → जबकि n 0 नहीं होता
        { → int q = n / 10;
          → int r = n % 10;
        }
    }
}
```

System.out.println (r);

$n = q$

}

HW Ques 2

## Rotate a number

number =

2 3 4 5 7

1 से rotate करना

दै तो last के 1 वे rotation + 1 =

7 2 3 4 5

digit के आगे लाना है।

rotation + 2 =

5 7 2 3 4

2 से rotate

बाकी है मतलब last के

2 digit के आगे लाना है।

3 से rotate करना है।

मतलब last rotation + 3 =

के 3 digit के आगे लाना है।

4 5 7 2 3

& so on

1 से rotate

2 से rotate

3, " " "

करना है मतलब 10 से divide करना है मतलब 100 से divide करना है

1000 " " "

2 से rotate करना। 97543 के तो 100 से divide करना।

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

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

no of digit in number

975 | 43

43975 → it answer  
पाइए

quotient = 975

remainder = 43

अब इसको 43975 बनाने के लिए

43 को 1000 से multiply करो और  
उसमे 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 =  $\text{let } K = 2 \text{ (10}^2\text{ Divisor)}$   
no. of digits in  $n = 6$

$$\begin{array}{r|rr} & 9 & 7 & 5 & 4 \\ \hline & 3 & 2 \end{array}$$

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

$$n-K = 6-2 = 4$$

$(10)^4$  = Multiplier

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

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

Divided by  $10^k$   
& None  
add  
the quotient  
with product  
of  $(10)^4$  multiplier  
& remainder

```

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();
        int nod = 0;
        int temp = num;
        while (temp != 0) {
            temp = temp / 10;
            nod++;
        }
        int div = 1;
        int mult = 1;
        for (int i = 1; i <= no. of digit; i++) {
            if (i <= k)
                div = div * 10;
            else
                mult = mult * 10;
            int q = n / div;
            int r = n % div;
            int ans = r * mult + q;
            System.out.println (ans);
        }
    }
}
  
```

*Important Side Cases*

$K = K/nod$   $\rightarrow$  if  $K = 25$   
 $K = 25 + 5 = 0$   
 $K = 98 - 47 + 5 = 4$   
 $\rightarrow$  equal EBT  $\rightarrow$  11  
 $\rightarrow$  num = 6753, rotation  
 $\rightarrow$  5th rot = 5th rot  
 $\rightarrow$  1st rotation = 611

|       |   |   |   |   |
|-------|---|---|---|---|
| 5     | 2 | 1 | = | 7 |
| digit | 3 | 4 |   | 8 |
| 4     | 1 |   |   | 9 |

Negative rotation  
कमज़ोर बाले दिग्गज पर्याप्त  
बाले बाले पर्याप्त

Negative Rotations में अंग्रे 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।