Email

- mohit.behl@pepcoding.com

Date ..../..../.....

flowchart

Flow → Design

10 Class → slow
- → Basic Syntax
- → Recursion

→ Extra Topic

DSA → Weekend Class

Regular Class

H.W discuss.

9 to 12

Regular - ~~9 to 12:00~~

Discuss
- → Resume building
- → Uses LinkedIn

M T W Th Fri Sat Sun

6-8:00 → Homework session.

(Homework will be discussed)

(MWF)

↓ turning

M-W-F

TTS → Regular Class

- 9 to 12.

Regular Concept / Question

→ H. Work

→ Discuss

→ Solution Video

→ Doubt Support

Notes
- → Approach
- → Code
- → Dry Run

Revision Roadmap

M T W Th Fri Sat Sun M T W Th Fri Sat Sun

Notes Revis

again after 1 week

Spiral

Date ..../..../.....

DSA using Java

A → B.

Output

System.out.println( ) ; → have to put

इसके बीच में जो Message लिखेंगे वह Print होगा।

Import [java.util.* ;] → need for our requirements

```
Import java.util *
public class Main {
public static void main ( String [] args ) {
    System.out.println ("welcome to my first class ;
                Buckle up for good coding exp.");
    }
}
```

next line

Sout → System को बोलो की output में Print करे जो " " में लिखा है।

Spiral

Date ..../..../.....

System.out.print (" ")

→ need for printing the next statement in next line.

⇒ print the output

System.out.println (" ").

→ need for printing ~~to~~ the ~~next line for statement~~ next statement in same line

⇒ print + move to the next line.

[\n] → special character

→ need for printing statement in next line.

⇒ follow the syntax Rules.

;

" "

S capital. ← System

Spiral

Date ..../..../.....

Output

→ variables → Java

Data store

A → int → Range small

number ← int (≠ decimal)

B ← char → a, b, c, @, #, $ size

true, false ← boolean → C

D [float, double] → decimal

A — long → works like int but has larger range

G — string → collection / sequence of character.

string = "pep coding" → always used in double quote.

Question

12345 → A → int

"123 ABC" → G → string

true → boolean

10.15 → Double / float

'#' → char

Char → used in single quote but only single symbol

String → used in double quote and use multiple things write

Spiral

Date ..../..../.....

'abc123' → error

↳ since string is used in " " and char is used in ' ' but single symbol

int v1 → container which stores only integers but can be named anything.

psvm {

int V1 ; // declaration

V1 = 100 ; // Initialization

(=) → assignment operator.

→ int V2 = 110 ;

Spiral

% → remainder

Date __/__/__

There is no concept of garbage value

```
int v1 ;  // declaration
v1 = 12 ;  // Initialization

int v2 = 10 ;
```

→ Arithmetic operator

```
int add = v1 + v2 ;
int sub = v1 - v2 ;
int mul = v1 * v2 ;
int div = v1 / v2 ;
int mod = v1 % v2 ;

cout ("v1 : " + v1 + " v2 : " + v2)
cout ("add : " + add) ;
cout ("sub : " + sub) ;
cout ("mul : " + mul) ;
cout ("mod : " mod) ;
```

+v2 → prints the value of v2

add : 22 mul : 120
sub : 2 mod : 2
div : 1

Output

v1 : 12 v2 : 10
add

Quotient
/ = 1.2 ≈ 1

- (rem)
% = 2
(rem)

Application of rem (%)
num → odd or even
num % 2 → 0 (even)
num % 2 → 1 (odd)

Spiral

Date ..../..../.....

→ last digit of a number
num % 10 → digit

Example 12345 % 10 → 5
123456 % 10 → 6

= → assignment operator

== → Equality operator (to check)

```
int v1 ;
v1 = 11 ;
int v2 = 10 ;
```

```
boolean isEven = (v1 % 2 == 0) ;
boolean check 1 = (v1 > v2) ;
boolean check 2 = v1 >= v2
boolean check 3 = v1 < v2
boolean check 4 = v1 <= 12
cout (isEven) ; → false
cout (check 1) ; → true
cout (check 2) ; → true
cout (check 3) ; → false
cout (check 4) ; → false
```

(||) → for true any one of the condition should be true.
↙ or operator

(&&) → for true both the conditions should be true.
← And operator

Raining	Bad weather	Need umbrella?
true	true	true
true	false	true
false	true	true
false	false	false

10th Pass	11th Pass	eligible for 12th
true	true	true
false	true	false
true	false	false
false	false	false

Spiral

Date ..../..../.....

logical operator

!= (not equal to)

true → false
false → true.

Assign

equality
True false

> → greater
>= → greater than equal to
< → less than
<= → less than equal to

Decision

Syntax

```
if (true) {
    ____
    ____
}
else {
    ____
    ____
}
```

Question

```
if (a>b && a>c) {
    sout ("a is larger");
} else if (b>a && b>c) {
    sout ("b is larger");
}
else if (c>a && c>b) {
    sout ("c is larger");
}
```


homework → grading system

Date ..../..../.....

a > b
True → a is largest so far
false → b is largest so false

b > c
True → b is largest
false → c is largest

a > c
True → a is largest
false → c is largest

a>b && a>c ⇒ a
a>b && c>a ⇒ c
b>a && b>c ⇒ b
b>a && c>b ⇒ c
largest

Code

```
a>b  if (a>b) {
         if (a>c) {
             sout ("a is largest");   a>c
         } else {
c>a          sout ("c is largest");
         }
     } else {
         if (b>c) {
             sout ("b is largest");   b>c
b>a  } else {
c>b      sout ("c is largest");
     }
     }
```

Date ..../..../.....

Note → approach
→ logic
→ little code part.

loops → Iterative method
↓
(बार बार)

for loop → Basic syntax

① starting point
② End condition / stopping condition
③ steps

```
for ( ; ; ) {
}
```

starting point → Initialized only one time
end condition → works like a lock
steps

```
for ( ; ; ) {
}
```

the thing which we're to print

Example

```
for (int i=1; i<=5; i++) {
    System.out.println("hello" + i);
}
```

i++ / (i+1) → value को एक से बढ़ाता है

साथ में i की value प्रिंट होगी

Output

hello 1
hello 2
hello 3
hello 4
hello 5

Spiral

(i++ → post Increment
++i → pre Increment)

Date ..../..../.....

Ques

```
for (int i=1; i<=10; i=i+3) {
    System.out.print(i+" ");
}
```

i = .

Initial value

1. for (int i=0; i<=10; i=i*1) {
   sout (i+" ");

→ infinite loop

→ ~~sari values zero till 10~~
infinite loop.

= 1, 4, 7, 10 → i

2. for (int j=1, i=5; j<=20; i=i+j, j=i) {
   sout (i+" ")

→ 5, 6, 12

Ans

i=i+j

i = ~~5~~ ~~6~~ ~~12~~ 24
j: ~~1~~ ~~6~~ ~~12~~ 24

3. int i=1;
   for (i=i+1; i<=10; i=i+2) {
   sout (i+" ");
   i=i-1;
   }

i = ~~1~~ ~~2~~ ~~3~~ ~~2~~ ~~4~~ ~~3~~ 5

→ 2, 4, 6, 8, 10

2, 3, 4, 5 - - 10.

Spiral

Date ..../..../.....

```
int sum = 0
    for (int i=1 ; i<=100 ; i++ ) {
        sum = sum + i ;
}
```

$\boxed{n(n+1)/2}$

Number ⇒ Input

```
java.util.Scanner ;
```

helps in taking Input

```
Scanner scn = new Scanner (System.in)
Int v = scn.nextInt ();
```

```
public static void main (String[] args) {
    Scanner scn = new Scanner(System.in);
    int b = scn.nextInt();
    System.out.println("value : "+val);
}
}
```

Spiral

Date ..../..../.....

```
Int n = in.nextInt();
Int count = 0
while (n!=0) {
    n=n/10
    count ++
}
Sout (count);
}
}
```

10	12345
10	1234
10	123
10	12
10	1
	0

n/10 → digits is decreased to 1

count = 0
1
2
3
4
5 → no. of digits

to count the no. of digits of a number entered by the user

~~public~~

```
psvm
Scanner sc = new Scanner (System.in);

int n = sc.nextInt();
    Int count = 0 ;

while (n!=0) {
    n = n/10 ;   // digit dec
    count ++ ;
}
```

or

```
for ( ; n!=0 ; ) {
    n = n/10 ;
    count ++ ;
}
System.out.println(count) ;
}
}
```

Spiral

Date ..../..../.....

```
Scanner sc = new Scanner(System.in)

int n = sc.nextInt();

int count = 0
while (n > 0) {
    int rem = n % 10;
    System.out.print(rem);
    n = n % 10; n = n/10;

}
}
}
```

```
int count = 0
while (n > 0) {
int rem = n % 10;
System.out.(rem)
n = n/10;
}
}
```

used for decreasing the no.

% → ~~to decrease the no. of digits~~
~~If we want to~~ ~~get~~ 1
अगर आखरी digit निकालना है
तो %

/ → divide → decrease the no. of digits.
↓
decrease.

```
n != 0
```

10	12345	
10	1234	5
10	123	4
10	12	3
10	1	2
	0	1

Spiral

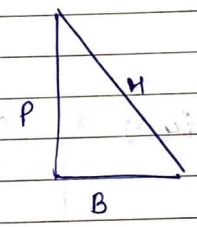
reverse a number

Date ..../..../.....

psvm

```
while (n != 0)
    int digit = num % 10;
    num = num / 10;
    sout (digit);
}
}
}
```

→ jab tak answer 0 nhi aa jaata



a, an ~~integer~~
b, an integer
c, an integer.

$(5)^2 = (3)^2 + (4)^2$

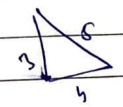
$25 = 9 + 9$

pythagorus triplet

If b > a and c

$b^2 = a^2 + c^2$

```
while (n != 10) {
    int count = 0;
    n = n/10; // digit dec
    count ++;
}
```



$a^2 = b^2 + c^2$

$= (3)^2 + (4)$

$=$

Spiral

Date ..../..../.....

Fibonacci Series

Eg.

0 1 1 2 3 5 8 13 21 34

F S

Sample Input

10 ⇒ n

Sample Output

0
1
1
2
3
5
8
13
21
34.

```
int n = scn.nextInt();
int first = 0, second = 1;
n times
for ( int i = 1 ; i <= n ; i = i + 1) {
    int third = first + second
    first = second ;
    second = third ;
}
```

i	first	second	third
1	0	1	1
2	1	1	2
3	1	2	3
4	2	3	5
5	3	5	8
6	5	8	13

Spiral

Date ..../..../.....

Prime number

↳ factors → 1 or number itself

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

boolean isprime = true;
for ( int div = 2 ; div <= n-1 ; div++) {
    if (n % div == 0) {
        // factor
        isprime = false ;
        break ;
    }
}

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

Date ..../..../.....

```
while (t!=0){

int n=scn.nextInt();

boolean isprime = true;
for (int div =2; div * div <= n ; div ++){

    if (n% div ==0){
      // factor
    is prime =false ;
    break;
  }

}

if (is prime) {
  System.out.println("prime");
}
  else {
System- out . println("not prime");
}
  t-- ;
}
```

Date ..../..../.....

```
Import java.util - Scanner ;
class A

{

public static void main (String [] args) {

  Int n, count = 0;
  System.out.print("Enter any number ");
  Scanner sc= new Scanner (System.in);
  ~~n = sc~~. n= nextInt ();

  for (int i=1; i<=n ; i++)
    {
      if (n% i ==0)
      {
        count ++ ;
      }
    }

    if (count == 2 ){
      System.out.println(" It is a prime no.");
  }
    else {
    System.out.println("not a prime no. ");
  }
}

}
```

Spiral

Date ..../..../.....

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

for (int i = 0; i<t ; i++) {

    int n = scn.nextInt();

    int count = 0;
    for (int div = 2 ; div*div <=n ; div++) {
        if (n % div == 0) {
            count++ ;
        }
    }

    if (count == 0) {
        System.out.println("prime");
    }
    else {
        System.out.println("not prime");
    }

}.
```

Spiral

Date ..../..../.....

(GCD/HCF and LCM)

$n_1 * n_2 = LCM * GCD$

$$\boxed{LCM = \frac{(n_1 * n_2)}{GCD}}$$

~~factor <= n~~ , → factor unique
18, 36

~~$\sqrt{18} \approx 4$~~

$\frac{n}{2}$ → किसी भी no. का factor होगा n/2 से less होगा

or min/2

```
int min=0
if (n1<n2)
    min=n1
else
    min=n2
```

① min = 18

② if (n1 % min == 0 && n2 % min == 0) {GCD = min}

③ for (i = min/2 ; i>=1 ; i--) {
    if (n1 % i == 0 && n2 % i == 0) {
        GCD = i
        break ;
    }
}

Spiral

Date ..../..../.....

जब तक दूसरा no. दूसरे no. को completely divide na करे

तब तक काम करते रहो।

$n_1 \% n_2$ → is divided by

छोटा no → n_1 → GCD.

छोटा no.

case (1) → $n_1 < n_2$ — बड़े no. से divide हो रहा है

$n_1 = 15$
$n_2 = 27$

rem = 15

Spiral

Date ..../..../.....

```
import java.util.*;

public class Main {

    public static void main (String[] args) {

        Scanner scn = new Scanner (System.in);

        int n1 = scn.nextInt(), n2 = scn.nextInt();

        int product = n1 * n2;

        while (n1 % n2 != 0) {

            int rem = n1 % n2;
                n1 = n2;
                n2 = rem;
        }

        int gcd = n2;
        int lcm = (product) / gcd;

        System.out.println (gcd);
        System.out.println (lcm);
    }
}
```

$n_1 < n_2$

15 n_2
n_1 27

n_1 = ~~15~~ ~~27~~ ~~15~~ 12
n_2 = ~~27~~ ~~15~~ ~~12~~

rem = 3

27)15(0
0
15)27
-15
12)15(1
-12
3)12(4
-12
0

Spiral

Doubt

Date ..../..../.....

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

Scanner scn = new Scanner (System.in);
int n = scn.nextInt();

for (int fac = 2; fac * fac <= n; fac ++) {

    while (n % fac == 0) {
    System.out.print (fac + " ");
        n = n/fac;
    }

}
if (n != 1) {
    System.out.println(n);
}
}
}
```

Spiral

Date ..../..../.....

fluctuation = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 10

B_1	off	on			
B_2	off	on	off		
B_3	off	on	off		
B_4	off	on	off	on	
B_5	off	on	off		
B_6	off	on	off	on	off
B_7	off	on	off		
B_8	off	on	off	on	off
B_9	off	on	off	on	
B_{10}	off	on	off	on	off

1
4
9

Observation

perfect square bulbs are on.

Spiral

Date ..../..../.....

```
import java.util.Scanner;
public class main {
    psvm (String args[])
{
    Scanner Scn = new Scanner(System.in);
    Int n = scn.next Int();
    for (Int i=1 ; i*i <=n ; i++) {
        System.out.println(i*i);
    }
}
}
```

1x36	36.
2x18	18x2
3x12	12x3
4x9	9x4
6x6	

k

non perfect square

= even factors

$\rightarrow 2^k$

perfect square

= odd factors

$\boxed{2^k + 1}$

↳ perfect square factors

even toggle = off non-perfect square = even factors

odd toggle = on perfect square = odd factors

Spiral

Date ..../..../.....

Inverse ≠ Reverse

Input

num = 283467 51

pos = 876543 21

digit → pos

~~283467 51~~

~~73~~

num = 15243

pos = 54321

num = 42135

pos = 54321

Output

num = 7 3 4 2 5 6 8 1

pos = 8 7 6 5 4 3 2 1

num = 1 8 9 3 4 7 6 5 2

9 8 7 6 5 4 3 2 1

num = 7 8 4 3 2 5 6 1 9

9 8 7 6 5 4 3 2 1

(digit, pos)

15243 = $(1\cdot10^4) + (5\cdot10^3) + (2\cdot10^2) + (4\cdot10^1) + (3\cdot10^0)$

↳ (digit × 10^{pos-1}) (notice)

42135 = $(4\cdot10^4) + (2\cdot10^3) + (1\cdot10^2) + (3\cdot10^1) + (5\cdot10^0)$

42135 = $(4\cdot10^4) + (2\cdot10^3) + (1\cdot10^2) + (3\cdot10^1) + (5\times10^0)$

↳ (pos × $10^{digit-1}$)

digit → pos

1×10^2

2×10^3

3×10^1

4×10^4

5×10^0

Spiral

Date ..../..../.....

Inverse of a number

psvm

```
Scanner sc = new scanner (System.in);
Int n = scn.nextInt();

Int ans = 0;
Int pos = 1;

while (n != 0) {

    Int digit = n % 10;
    ans += (pos * Math.pow(10, digit-1));

    n = n/10;
    pos ++;

}

Sout (ans);
```

10	1 5 2 4 3
10	1 5 2 4
10	1 5 2
10	1 5
10	1
	0

digit	pos
3	1
4	2
2	3
5	4
1	5

$ans = ans + (pos * 10^{digit-1})$

$ans = (1 \cdot 10^2) + (2 \cdot 10^3) + (3 \cdot 10^1) + (4 \cdot 10^4) + (5 \cdot 10^0)$

$= 100 + 2000 + 30 + 40000 + 5$

$= 42135$

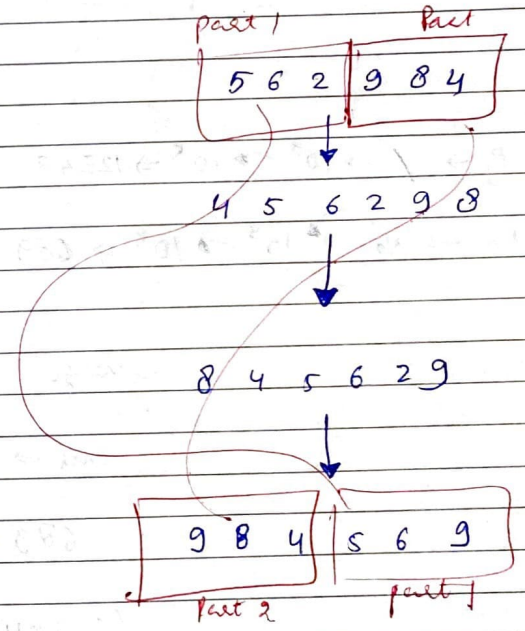
Date ..../..../.....

Rotation of a number

(+ve Rotation)
पीछे से एक digit आगे आया

Sample Input

5 6 2 9 8 4



Input → 562 984

→ (562 × 1000) + 982

Date ..../..../.....

$\boxed{K=3}$

① split

Input ⇒ count - K K → pow → 10^K ⇒ 10^3

12 5 4 3 6 8 9

P_1 P_2

P_1 → / → 10^3 → 10^K ⇒ 12543

P_2 → % → 10^3 → 10^K ⇒ 689

merge

ans ⇒ $P_2 P_1$

689 12 5 43

$(689 \times 10^5) + (12543)$

ans ⇒ $(P_2 \times 10^{count - K}) + P_1$

जब ज़बरदस्ती चीज़ निकालनी हो तो आगे int लगा दो।

Spiral

Date ..../..../.....

हमेशा 10^3 से divide करना है

e.g 1 2 3 4 5 6 7 8

$P_1/10$ $P_2\%10$

12345

⇒ $\boxed{678\ 12345}$

```
Scanner Scn = new Scanner(System.in)

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

// digits count

int count = 0;
int temp = n;

while (temp != 0) {

    temp = temp/10
    count ++;

// split
int splitHelper = (int) Math.pow(10, K);
int p1 = n/splitHelper;
int p2 = n%splitHelper;
```

12543

K=3

count = 5 12345 678/100

P_1 P_2

Split helper = 10^3 = 1000

P_1 ⇒ 12

P_2 ⇒ 543

merge helper = 10^2 = 100

⇒ $(543 \times 100) + 12$

ans → 54312

Spiral

7 April Thursday → Leave

~~Next week Monday, Saturday & class~~

(Sunday)

26th April = Tuesday → 10th April (6:00 - 9:00) Pm

reschedul

28th April = Thursday → 17th April (Sunday)

reschedule [12:00 - 3:00] Afternoon.

30th April - Saturday

Test

// merge

```
int mergeHelper = (int) Math.pow (10, count - K);
int ans         = (p2 * merge helper) + p1;

System.out.print ln (ans);
```