

28/4/16

$$(322)_8 \rightarrow (110001000)_2 \rightarrow (188)_{16}$$

322 → Decimal

Binary →

512	256	128	64	32	16	8	4	2	1
1	1	0	0	0	1	0	0	0	0

$$(322)_{10} \rightarrow (110001000)_2$$

octal →

Take Binary num and group it to 3

1	1	0	0	0	1	0	0	0
4	2	1	4	2	1	4	2	1
6	1	0						

Why group it to 3?
Octal num from 0 to 7

$$(322)_{10} \rightarrow (110001000)_2 \rightarrow (610)_8$$

hexadecimal →

Take Binary Equallent Num & group it to 4

0	0	0	1	1	0	0	0	1	0	0
8	4	2	1	8	4	2	1	8	4	2

1 8 8

$$\Rightarrow (188)_{16}$$

$$(322)_{10} \rightarrow (110001000)_2 \rightarrow (188)_{16}$$

$(356)_8$

Binary \rightarrow

3	5	6
$4^2 + 4^2 + 4^2$	$4^2 + 4^2 + 4^2$	$4^2 + 4^2 + 4^2$
1	1	1

 $(011101110)_2$

$$(356)_8 \rightarrow (011101110)_2 \leftarrow \text{of } 2^2 \text{ s}$$

Decimal \rightarrow

0	1	1	1	0	1	1	1	0
256	128	64	32	16	8	4	2	1

$$(238)_{10} = 10010110$$

$$128 + 64 + 32 + 8 + 4 + 2 = 238$$

$$(356)_8 \rightarrow (238)_{10}$$

Hexa decimal \rightarrow

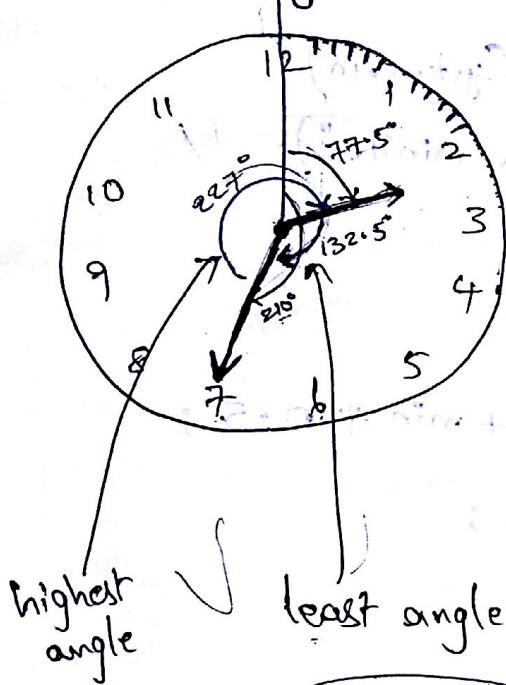
$$(011101110)_2 \rightarrow (EE)_{16}$$

$$(356)_8 \rightarrow (011101110)_2 \rightarrow (EE)_{16}$$

$(AB)_{16}$

1	1	1	1	0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1	1	0	0	0

Q) WAP to find the angle b/w hour Arm & minute Arm in Analog clock.



Total $\rightarrow 360^\circ$ and $\frac{360}{12} \Rightarrow 30^\circ$
 Every hour \rightarrow Rotate 30°
 Every min \rightarrow moves 0.5°
 $60 \text{ min} \rightarrow 30^\circ$
 $1 \text{ min} \rightarrow 0.5^\circ$
 $\frac{30}{60} = \frac{1}{2} = 0.5^\circ$

* Take Every time least angle

$$\begin{aligned}
 &\text{2 hr} \rightarrow 2 \times 30^\circ + 35 \times 0.5^\circ \\
 &\rightarrow 60^\circ + 17.5^\circ \\
 &\rightarrow 77.5^\circ \\
 &\text{hour hand moves} \\
 &\text{35 min} \\
 &\rightarrow \text{minute hand moves} \\
 &35 \times 6^\circ \rightarrow 210^\circ
 \end{aligned}$$

For Every min
hour hand moves
 0.5° with minute hand

$$\begin{aligned}
 &360^\circ \Rightarrow 6^\circ \\
 &1 \text{ min} \rightarrow 6^\circ \Rightarrow 1 \text{ min} \rightarrow 6^\circ \Rightarrow 1^\circ \\
 &\frac{360}{60} = 6^\circ
 \end{aligned}$$

$$77.5 + 210 \rightarrow 287.5$$

$$\begin{aligned}
 &210 \\
 &77.5 \\
 &\text{least angle} \rightarrow 132.5 \\
 &132.5 < 180^\circ \\
 &287.5 \Rightarrow \text{highest angle} \\
 &287.5 > 180^\circ
 \end{aligned}$$

class Clock

{
 psvm()

{

Scanner sc = new Scanner (System.in)

SOP ("Enter hour and minute");

int hr = sc.nextInt();

int min = sc.nextInt();

double hangle = hr * 30 + min * 0.5;

double mangle = min * 6;

double angle = 0.0;

if (hangle > mangle)

angle = hangle - mangle;

else

angle = mangle - hangle

if (angle > 180)

angle = 360 - angle;

SOP ("last angle : " + angle);

* WAP to print the multiplication table of given number

int n=3;

for (i=3; i<=n; i++)

for (j=1; j<=10; j++)

SOP ("i + " * + j + " = " + (i*j));
(n*j));

possible ways to write

for(; cond ; Exp);

for(; ;); → Infinite loop (Defaultly condition is true)

for(; ; Exp);

for(ini ; cond ;);

Note :- In output stmt SOP first

$\boxed{+}$ is more important than $\boxed{*}$
highest priority

④ WAP to display Sanju if num is multiple
of 3 else display geeta if num is multiple of 5

or display Sanju, geeta if it is multiple of
3 & 5 both

$n = \text{sc.nextInt();}$

if ($n \% 3 == 0 \& \& n \% 5 == 0$)

SOP("Sanju" ~~and~~ "geeta");

else if ($n \% 5 == 0$)

SOP("geeta");

else if ($n \% 3 == 0$ ~~& & $n \% 5 == 0$~~)

SOP("Sanju" ~~and~~ "Geeta");

29/4/16

Question of your choice

* Factorial of given Num

```
n = sc.nextInt();  
int fact = 1;  
while (n != 0) || while (n >= 1)  
{  
    fact = fact * n;  
    n--;  
}  
Sop ("fact of num is " + fact);
```

* Factorial of given 2 numbers

```
class main()  
{  
    static int fact (int n)  
    {  
        int fact = 1;  
        while (n >= 1)  
        {  
            fact = fact * n;  
            n--;  
        }  
        return fact;  
    }  
    class Mainclass  
    {  
        psvm()  
        {  
            for (int i=1; i<=10; i++)  
                Sop ("factorial of " + i + " is " + fact(i));  
        }  
    }  
}
```

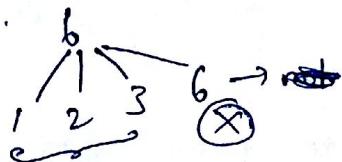
Given num is perfect or not



Sum of its divisors is equal to that num.

∴ The number itself also a one divisor. But we

will not consider that one]



$$1+2+3 = 6$$

class Perfect

{ sum = 0; perf = 0;

static int perf(int n)

{ int perf = 0;

for (i=1; i <= n/2; i++)

{ if (n % i == 0)

{ if (n % i == 0) {

perf = perf + i;

} }

return perf;

import java.util.Scanner;

class Mainclass

{ public static void main(String args[])

PSVM();

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

n = sc.nextInt();

int i = Perfect.perf(n);

if (i == n);

{ Sop("n + "is a perfect num");

else

Sop("n + "is not perfect num");

~~2/5/16~~

Prime

Given num is prime or not

Stigiphis laevigatus (L.) Linnaeus do. small

Class Main class test at lang3 error

6 static int isprime(int n) { } static boolean isprime(int n)

int temp = 0; (in for loop initial)

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

($a < \infty$) stable

if ($n \% i == 0$)

.temp = 13

(as yet) ~~it~~ lame

if (temp == 0)

return times
else

~~metam~~ false; ~~bookend~~

Public static void main (String[] args)

SOP ("---"):

```
int p = isprime(5); // boolean p = isprime(5);
```

if ($P == 0$)

```
if(isprime(5));
```

$\text{Sop}(\text{"}\phi\text{'' num is prime num"}) \rightarrow (\)$ result

else \rightarrow next file

→ Wet film

$\text{Sop}(\text{"num is not a prime"}) \rightarrow \text{IR} = 1$

Prime → 0 to 100 & Count & Sum of prime no's

(Calculator) write

class MainClass

{ static boolean isprime (int n)

{ int temp = 0; if (n < 2) return

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

{ if (n % i == 0)

{ temp = 1; break; }

} if (temp == 0)

return true;

else return false;

public static void main (String [] args)

(System.out.println()); int sum=0; int count=0;

for (int i=2; i<=100; i++)

{ if (isprime (i))

sum = sum + i;

count++; }

System.out.println ("Sum of prime no's b/w 0-100 is " + sum);

System.out.println ("no of prime no's b/w 0-100 is " + count);

}

```

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

```

```

for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        sop(*);

```

sopln();

```

for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        sop(i);

```

```

(sop(i) i -> i : s(i)) i
sopln();

```

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

```

for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        sop(i);

```

sopln();

1
2
3
4
5
6
7
8
9
10

```

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

```

(i -> i : s(i)) i

```

char ch;
for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        ch = 'a';

```

~~char ch;~~

sop(ch++); // sop((char)ch++);

sopln();

(i -> i : s(i)) i

Ascii values

a
b
c
d
e
f
g
h
i
j
k
l
m
n
o

char ch = 'a';

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

for (int j=1; j<=i; j++)

sop((char)ch++);

sopln();

```

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

```

```
for (i=1; i<=5; i++)
```

```
{ for (j=1; j<=5; j++)
```

```
if (i==1 || j==1 || i==5 || j==5)
```

```
    Sop("*");
```

(1,1)	*	*	*	*	*	(1,5)
(2,1)	*	*	*	*	*	(2,5)
(3,1)	*	*	*	*	*	(3,5)
(4,1)	*	*	*	*	*	(4,5)
(5,1)	*	*	*	*	*	(5,5)

```
for (i=1; i<=5; i++)
```

```
{ for (j=1; j<=5; j++)
```

```
if (i==j || i+j==6) || i==1 || j==1 ||
```

```
    Sop("*");
```

(++) Sopn();

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

```
    for (j=1; j<=n; j++)
```

```
if (i==1 || j==1 || i==5 || j==5 || i==j || i+j==n+1)
```

```
Sop("*");
```

(++) Sopn();

```

*   *
*   *
*   *
*   *

```

```
for (i=1; i<=5; i++)
```

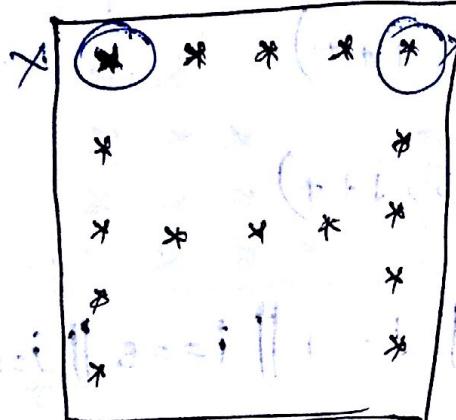
```
    for (j=1; j<=5; j++)
```

```
if (i==4 || (i==1 & j==3) || (i==2 & j==2) ||
```

```
(i==2 & j==4) || (i==3 & j==1) || (i==3 & j==5)
```

```
|| (i==5 & j==0) || (i==5 & j==5))
```

```
Sop("*");
```



$\Rightarrow \text{int } n=13$

$\text{for } (\text{int } i=0; i < n; i++)$

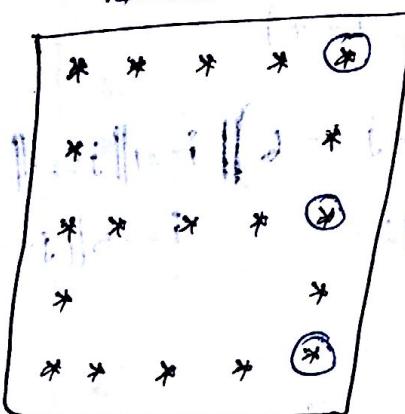
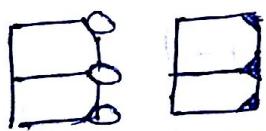
$\text{for } (\text{int } j=0; j <= n/2; j++)$

$\text{if } (i==0 \&& j!=0 \&& j!=n/2) \&$

$j==0 \&& i!=0 \&& j=n/2 \&& i!=0 \&$

$i=n/2)$

$\text{sop} ("*");$

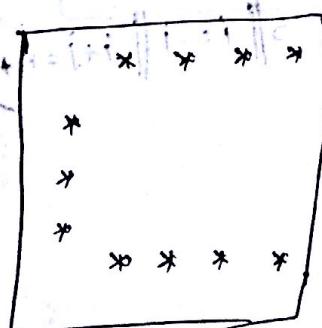


$\text{int, } n=13;$

$\text{for } (\text{int } i=0; i < n; i++)$

$\text{for } (j=0; j <= n/2; j++)$

$\text{if } ((i==0) \&& (j>i)) \&&$



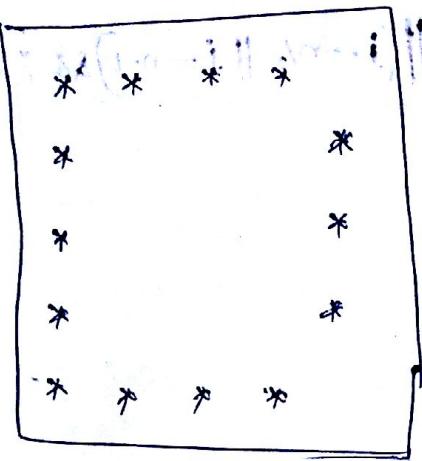
$\text{if } ((i==0) \&& (j==n-1)) \&& j!=0 \&& i!=0 \&&$

$i!=n-1)$

$\text{sop} ("*");$

else

$\text{sop} ("*");$

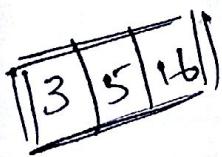


$\Rightarrow i < 0 \& (i > n-1) \& (j > 0) \& (j < n)$

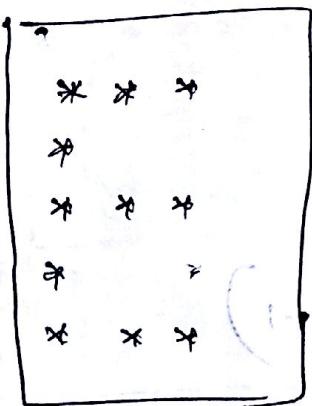
~~$f_t((i=0 \& i=n-1) \& j \neq n/2) \& j=0$~~

$j = n/2 \& i \neq 0 \& i \neq n-1$

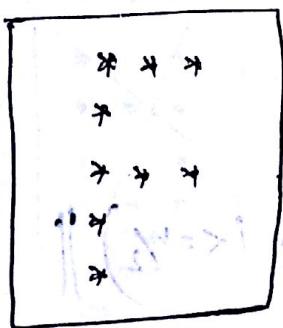
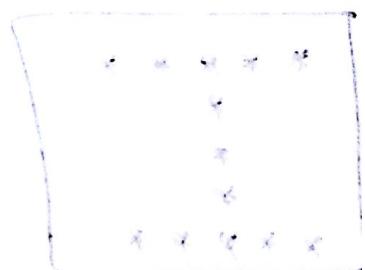
$i() \text{ else}$



$(i < 0 \& i > n-1) \& (j > 0 \& j < n)$

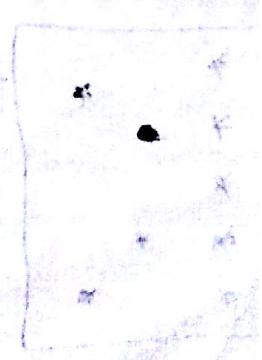
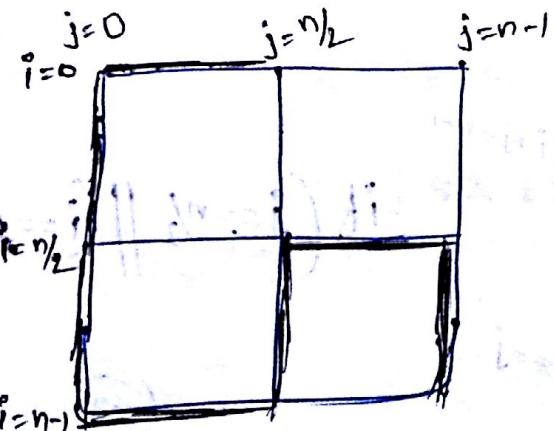
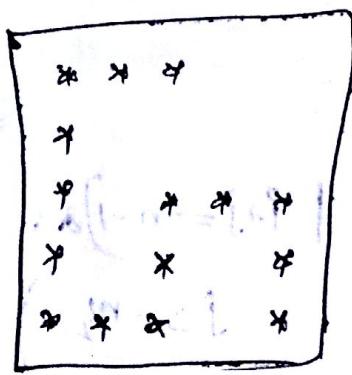
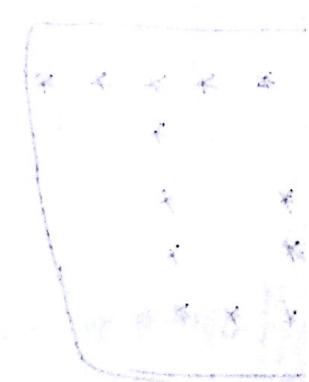


$f_t(i == 0 \& j == 0 \& i == n/2 \& i == n-1)$



$f_t(i == 0 \& j == 0 \& i == n/2)$

$\Rightarrow i < 0 \& (i > n-1) \& (j > 0 \& j < n)$



$\text{if } (j == 0 \parallel (i == 0 \parallel i == n - 1) \& \& j \leq n/2 \parallel (j == n/2 \parallel j == n - 1) \& \& i > n/2$
 $\parallel (i == n/2 \& \& j \geq n/2)$
 $\text{Sop}(\text{"*"});$
sopln();

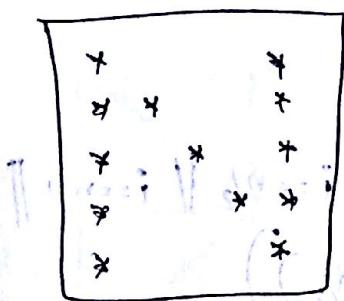
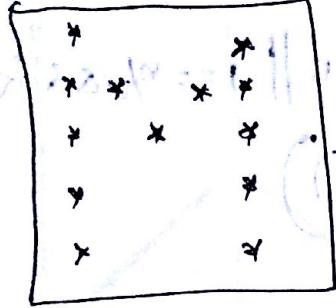
$\text{if } (j == 0 \parallel i == n/2 \parallel j == n - 1)$

$\text{if } (i == 0 \parallel j == n/2 \parallel r == n - 1)$

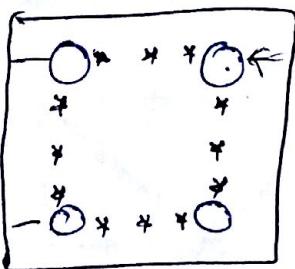
$\text{if } (i == 0 \parallel j == n/2 \parallel (i == n - 1 \& \& i \leq n/2) \parallel (j == 0 \& \& j \geq n/2))$

$\text{if } (j == n/2 \parallel (i == j \parallel i + j == n - 1) \& \& j > n/2)$

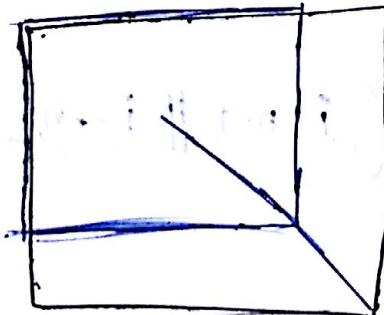
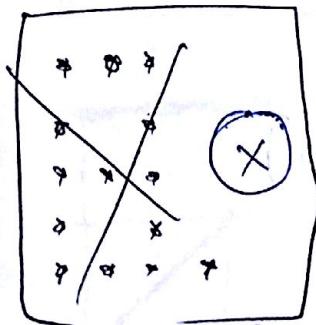
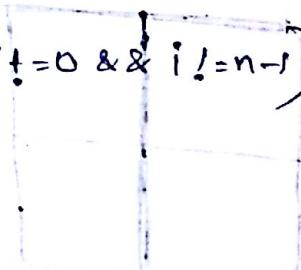
(69)
 $\text{if } (j == 0 \parallel (i == j \parallel i + j == n - 1) \& \& j \leq n/2)$



$\text{if } (j == 0 \text{ || } j == n - 1 \text{ || } \cancel{(i == j)})$



$\text{if } ((i == 0 \text{ || } i == n - 1) \& \& j != 0 \text{ & } j != n/2 \text{ || } (j == 0 \text{ || } j == n/2) \& \& i != 0 \text{ & } i != n - 1)$



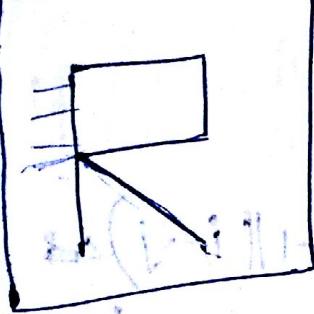
$\text{if } ((i == 0 \text{ || } i <= 3 * n / 4) \& \& j <= 3 * n / 4 \text{ || } (j == 0 \text{ || } j >= 3 * n / 4) \& \& i <= 3 * n / 4 \text{ || } i == j \text{ & } i >= n / 2)$

Sop ("*");

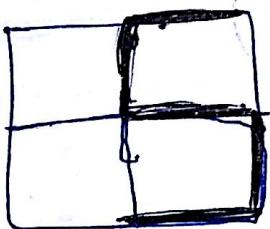
Sopln();



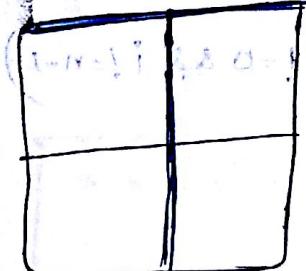
$\text{ff} (i == 0 \parallel i == n/2 \parallel j == 0 \parallel j == n/2 \&& i <= n/2)$



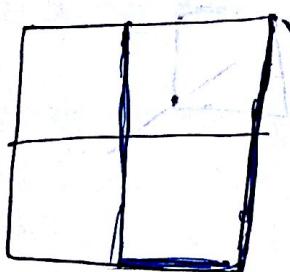
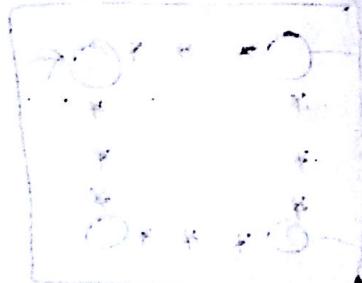
$i == n/2 \&& j <= n$



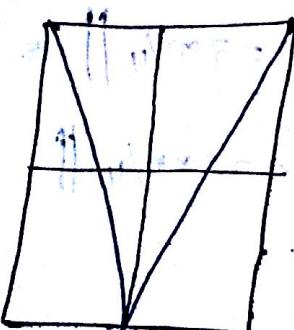
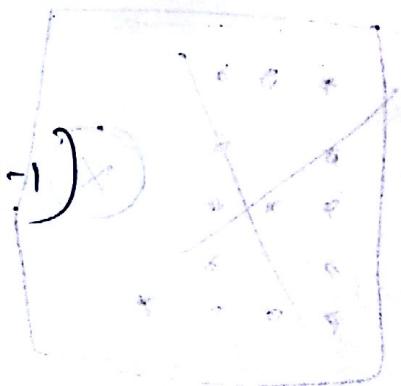
$\text{ff} (i == 0 \parallel (j == n/2 \&& i <= n/2) \parallel i == n/2 \parallel i == n-1 \parallel (j == n-1 \&& i >= n/2))$



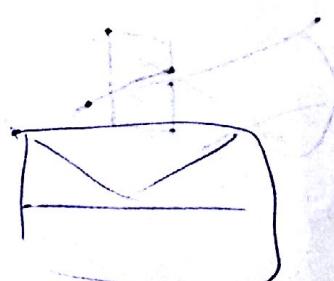
$\text{ff} (i == 0 \parallel j == n/2)$

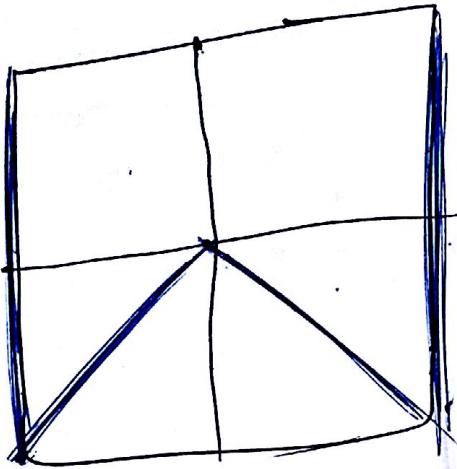


$\text{ff} (i == n-1 \parallel j == n/2 \parallel j == n-1)$



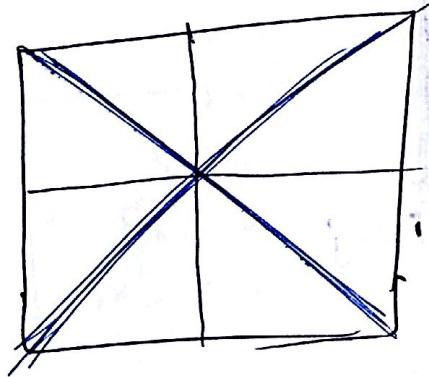
$\text{ff} (i == j \parallel (i+j == n-1) \parallel (i+j < i \&& j < i))$



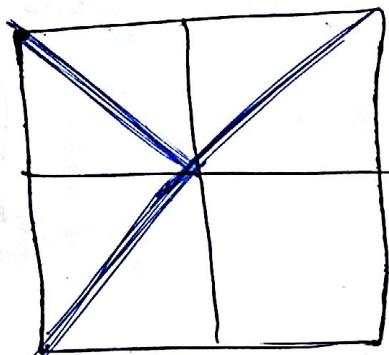


$\text{ff} (j == 0 \parallel j == n - 1)$

$((i == j) \parallel i + j == n - 1) \& \& i >= n/2$

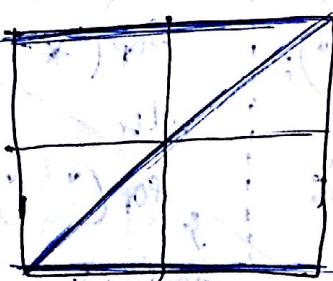


$\text{ff} (i == j \parallel i + j == n - 1)$

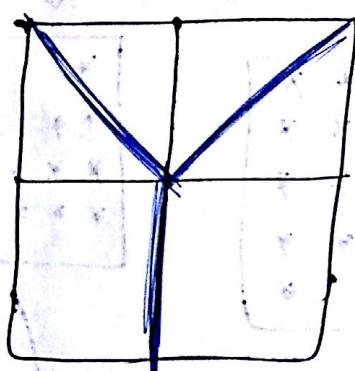


$\text{ff} ((i == j \& \& i <= n/2) \parallel i + j == n - 1)$

$(i + j == n - 1 \& \& i >= n/2)$

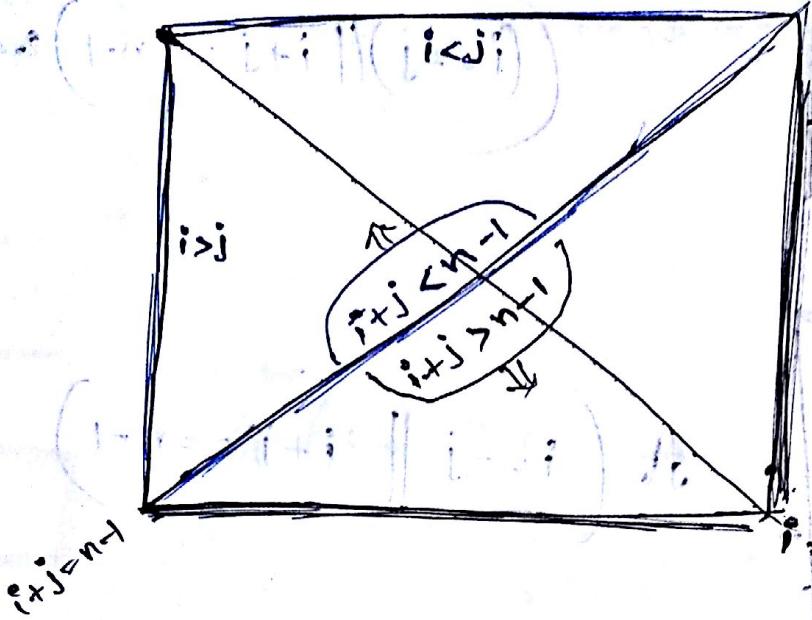


$\text{ff} (i == 0 \parallel i == n - 1 \parallel i + j == n - 1)$



$\text{ff} ((i == j \parallel i + j == n - 1) \& \& i <= n/2)$

$(j == n/2 \& \& i >= n/2)$



int n=13;

```
for ( int i=0 ; i<n ; i++ )
```

```
    for ( int j=0 ; j<n ; j++ )
```

{

```
    if ( i+j > n-1 )
```

```
        sop( " * " );
```

else

```
        sop( "   " );
```

sopln();

```
    if ( i < j )
```

```
        sop( " * " );
```

else

```
        sop( "   " );
```

sopln();

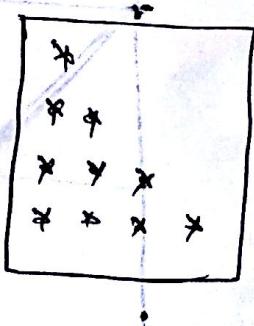
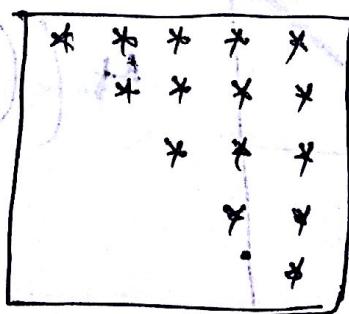
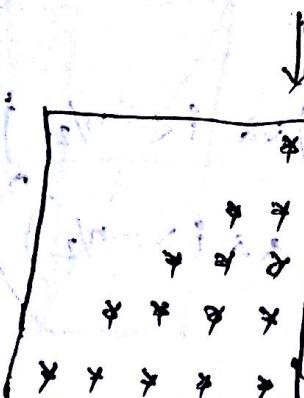
```
    if ( i > j )
```

```
        sop( " * " );
```

else

```
        sop( "   " );
```

sopln();



```

    --- *
    -- * *
    - * * *
    * * * * *

```

```

for (i=0; i<n; i++)
    for (j=0; j<n-1-i; j++)
        sop(" .");
    for (j=0; j<=2*i; j++)
        sop("* *");

```

```

    i=0
    j=0   1
           |
           2
    j=0   1   2
           |   |
           3   4
    j=0   1   2   3
           |   |   |
           4   3   2   1

```

printed result will be for for loop iteration

where j value odd \rightarrow print *

(*)

j value Even \rightarrow print start with 1

```
for (i=0; i<n; i++)
```

```
{ for (j=0; j<n-1-i; j++)
```

```
sop(" .");
```

```
int k=1;
```

```
for (int j=0; j<=2*i; j++)
```

```
{ if (j%2 != 0)
```

```
sop("* *");
```

```
else
```

```
sop(K++);
```

starting iteration of inner loop will be 1

starting iteration of inner loop will be 1

printing iteration outside, it is 1

```

1
0 1
0 1 0
0 1 0 1
0 1 0 1 0

```

At Even position \rightarrow 0

At Odd position \rightarrow 1

```
int count=1;
```

```
for (i=0; i<n; i++)
```

```
{ for (j=0; j<=i; j++)
```

```
sop((i+1)+count%2);
```

```
count++;
```

WAP to check

ARMSTRONG

→ Sum of power of ~~num~~

~~no~~ no. of digits of individual
digits is called ARMSTRONG

Ex:-

$$\textcircled{1} \quad 23 \neq 2^2 + 3^2$$

$$\textcircled{2} \quad 321 \neq 3^3 + 2^3 + 1^3$$

$$\textcircled{3} \quad 1234 \Rightarrow 1^4 + 2^4 + 3^4 + 4^4$$

$$\textcircled{4} \quad 12345 \Rightarrow 1^5 + 2^5 + 3^5 + 4^5 + 5^5$$

$$\textcircled{5} \quad 123456 \Rightarrow 1^6 + 2^6 + 3^6 + 4^6 + 5^6 + 6^6$$

2 is Armstrong $(2^1) = 2$

343 is Armstrong.

$$(3^3 + 4^3 + 3^3) = 343$$

ans
num/10 - 2
num/10 - 2
count++

```

class Armstrong {
    static int IsArmstrong(int n, int count) {
        int sum = 0;
        while (n > 0) {
            n = n % 10;
            sum = sum + pow(n, count);
            n = n / 10;
        }
        return sum;
    }
}

```

Public static void main (String[] args)

```

{
    System.out.println("Enter the num");
    Scanner sc1 = new Scanner(System.in);
    int num = sc1.nextInt();
    Armstrong(num) int temp = num;
    while (num > 0) {
        num = num % 10;
        count++;
        num = num / 10;
    }
    int g1 = IsArmstrong(temp, count);
    if (g1 == temp)
        System.out.println("temp + " + temp + " is a Armstrong number");
    else
        System.out.println("temp + " + temp + " is not a Armstrong number");
}

```

```

Static int pow (int n, int p) {
    int pro = 1;
}

```

while ($P > 0$)

{

$pro = pro * n;$

$P--$

getchar pro;

}

```

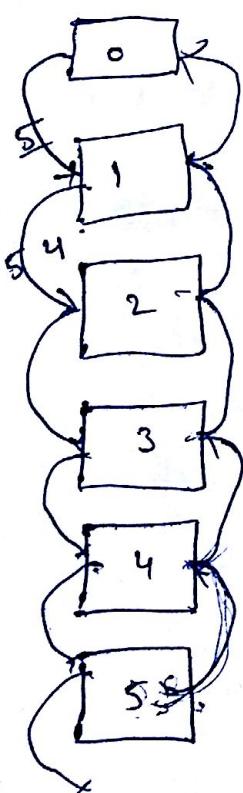
int n=3;
for (i=0; i<n; i++)
{
    for (j=0; j<n-1; j++)
        sop(" ");
    for (j=0; j<=i; j++)
        if (j==0)
            sop("*");
        else
            sop(" ");
    for (k=0; k<i-1; k++)
        if (k==i-1)
            sop("*");
        else
            sop(" ");
    sopln();
}
for (i=n-1; i>0; i--)
{
    for (j=0; j<n-i; j++)
        sop(" ");
    for (j=0; j<=i-1; j++)
        if (j==0)
            sop("*");
        else
            sop(" ");
    for (k=0; k<i-1; k++)
        if (k==i-1)
            sop("*");
        else
            sop(" ");
    sopln();
}

```

(a) (i) think blow step
(a) (ii) ~~if~~

WAP to print 1 to 10 with out using a loop?
(a) ~~using~~

```
class Main
{
    static void value(int n)
    {
        SOP(n);
        PSUM();
        return value(n-1);
    }
}
```



```
class Main
{
    static void value(int n)
    {
        SOP(n); if(n!=0)
        int r=get(value(n-1));
        SOP(n);
    }
}
```

int n=~~get~~ value(10);
SOP(n);

if(n!=0) { r = ~~get~~ value(n-1); }
else { r = 0; }

Recursion :- A function calling itself is called Recursion.

- ① WAP to print 1 to 10 numbers without using loop.

class Main

{

 static void print(int n)

{

 if (n > 0)

 print(n - 1);

 System.out.println(n);

 }

}

 public static void main(String[] args)

{

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

 print(10);

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

- ② WAP factorial of a given num Using Recursion

class Main

{

 static int factorial(int n)

 { int fact = 1; do { fact = fact * n; }

 n = n - 1; while (n > 0);

 } fact = fact * factorial(n - 1);

 return fact;

```

class main {
    static int fact (int n) {
        if (n == 0) return 1; // base condition
        return n * fact (n-1); // recursive call
    }
    psum()
    {
        int f = fact (5);
        sop (f);
    }
}

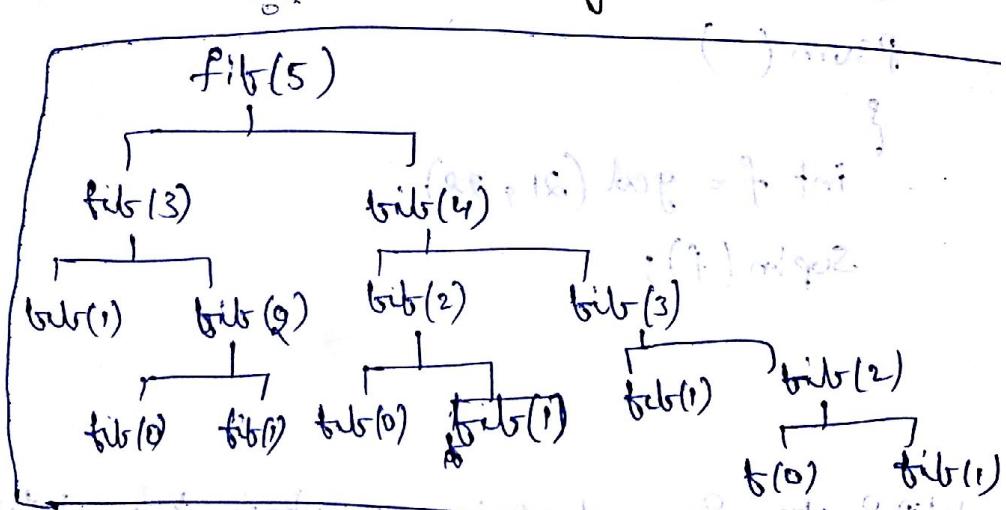
```

1* fact(0) = 1
 2* fact(1) ← $2 \times 1 = 2$
 3* fact(2) ← $3 \times 2 = 6$
 4* fact(3) ← $4 \times 6 = 24$
 5* fact(4) ← $5 \times 24 = 120$
 fact(5) ← 120

(n=0, n=1) base condition
 (n=2, n=3) recursive
 (n=4, n=5)

program complete

wap to fibonacci numbers using recursion.



```

class main {
    static int fibi (int n) {
        if (n == 0)
            return 0;
        else if (n == 1)
            return 1;
        else
            return fibi(n-1) + fibi(n-2);
    }
    psumc()
    {
        int f = fibi(8);
        sop (f);
    }
}

```

~~fibi(n-1) + fibi(n-2)~~

Q) WAP to find G.C.D of a number
 (a) Take two numbers(m, n) from the user
 (b) Swap
 (c) If m > n then
 class main()
 {
 static int gcd(int m, int n)
 {
 if (m < n)
 return gcd(n, m);
 if (n == 0)
 return m;
 return gcd(n, m % n);
 }
 psum()
 {
 int f = gcd(21, 28);
 System.out.println(f);
 }
 }
 WAP to Sum of prime numbers b/n 1 to 100
 WAP to find the Sum of multiple of 3 and 5 b/n 1 to 100

a
 b
 c
 d
 e

```

    int i, j, K, n = 5;
    char ch = 'a';
    for (i=0; i<n; i++)
    {
      for (j=0; j<=i; j++)
        SOP((char)(ch+i) + " ");
      K = K + n - i - j;
    }
    SOP(K);
  
```

Sum of prime numbers

```

class main
{

```

```
  static int prime(int n)
```

```

  {
    int temp = 0;
    int sum = 0;
  
```

```
    for (int i=2; i<=n; i++)
  
```

```

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

```

    {
      if (i%j == 0)
    
```

```

      temp = 1;
    
```

```
    break;
  
```

```

  if (temp == 0)
  
```

```

    sum = sum + i;
  
```

```

  return sum;
}

```

```

Public static void main (String [] args)
{
    Sop ("---");
    Scanner sc1 = new Scanner (System.in);
    i ("");
    Sop ("Enter no. of prime numbers you want to sum");
    int num = sc1.nextInt ();
    int sum = prime (num);
    Sop ("Sum of prime no's b/w 1 and " + num +
        " is " + sum);
}

```

Sum of multiple 3 & 5

class main

```

static int summul (int n1, int n2)
{
    int sum=0;
    for (i=n1; i<=n2; i++)
    {
        if ((i%3==0) && (i%5==0))
            sum = sum + i;
    }
    return sum;
}

```

```

Public static void main (String [] args)
{
    Sop ("---");
    Sop ("Enter starting num");
    int num1 = sc1.nextInt ();
    Sop ("Enter upto which num");
    int num2 = sc1.nextInt ();
}

```

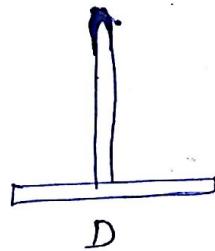
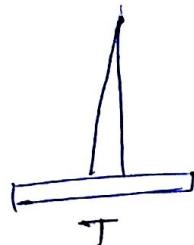
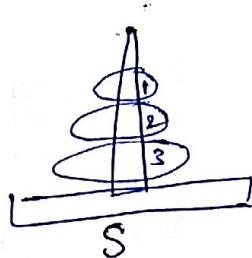
`int Sum = Sum-mul (num1, num2);`

`SOP ("Sum of multiples of 3 & 5, in bin" + num1 +
and " + num2 " is " + Sum);`

}

9/5/15

BY USING TOWER OF HENNAI



- 1 move S to D
- 2 move S to T
- 1 move D to T
- 3 move S to D
- 1 move T to S
- 2 move T to D
- 1 move S to D

④ for n numbers of plates

④ $2^n - 1$ moves

\downarrow *using 3 plates*

(note: each move, need moves + 1) $2^n - 1 \Rightarrow 2^3 - 1 = 7$ moves

Void move-Disc (*for base case*)
{
 if ($n == 1$) {
 cout << "moving plate 1 from " + src + " to " + des;
 }
}

`Sopln (n + " moving from " + src + " to " + des);`

`return;`

`move-Disc (n-1, src, des, Temp);`

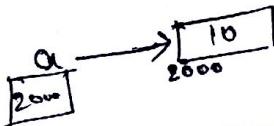
`Sopln (n + " moving from " + src + " to " + des);`

`move-disc (n-1, Temp, src, des);`

}

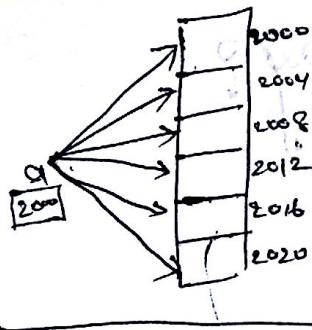
ARRAYS

`int a = 10`



Here 'a' is an identifier and it is pointing to one memory location.

`int a[]`



Here 'a' is an identifier and it is pointing to more than one memory location and also we make that a variable pointing to one memory location.

`a[0], a[1], a[2], a[3], a[4]`

`a[5] (X)`

`<datatype> <identifier> [] ;` // Array Declaration

`int a[];`

`double d[];`

`<identifier> = new <datatype> [size];` // Allocation

`a = new int [5]`

`d = new double [4];`

`<identifier> [index] = Values;` // Initialisation

~~Index~~ * \Rightarrow index \leq size

`a[2] = 21;` ✓

`a[-1] = 35;` X at compile time

`a[5] = 45;` X at runtime

Declaration + Allocation

`<datatype><identifier>[] = new <datatype> [Size];`
`int a[] = new int[5];` (size + array)
`double d[] = new double[4];`

Declaration + Allocation + Initialization

`int arr[] = {10, 20, 30, 40, 50}; // d/A;`
`int mark[] = {95, 96, 97, 98, 99}; // d/A;`

How to get how many elements in an Array?

→ mark.length; is returned
 How to get Size of datatype?

`Size = sizeof(datatype);`

10/5/16 Sum of Even numbers & Odd numbers in a Array

```
import java.util.Scanner;
class main {
    psum()
    {
        Scanner sc = new Scanner(System.in);
        Sop("Enter the number of elements");
    }
}
```

```

int n = sc.nextInt(); // & happier
Sop("Enter " + n + " elements"); // input
int arr[] = new int[n]; // information about
for (int i = 0; i < n; i++) // loop
{
    arr[i] = sc.nextInt(); // input
}
int evenSum = 0, oddSum = 0; // EC=0, OC=0
for (int i = 0; i < n; i++) // loop
{
    Sop(i + " ---> " + arr[i]); // output
    if (arr[i] % 2 == 0) // condition
    {
        evenSum += arr[i]; // addition
    }
    else // condition
    {
        oddSum += arr[i]; // addition
    }
}
Sop("Even numbers Sum " + evenSum); // output
Sop("Odd numbers Sum " + oddSum); // output
}

```

* 1st Biggest & 2nd Biggest Element from a Given Array (With out Sorting)

class Main

{
PSUM()

int arr[] = {12, 34, 32, 54, 45, 6, 17};

int n = arr.length;

System.out.println(" number of elements : " + n);

int fbig = arr[0];

int sbig = arr[0];

for (int i = 1; i < arr.length; i++) { int fbig = 0; int sbig = 0;

if (arr[i] > fbig)

sbig = fbig;

fbig = arr[i];

else if (arr[i] > sbig)

sbig = arr[i];

SOP("1st biggest element in array is " + fbig);

SOP("2nd biggest element in array is " + sbig);

~~fbig = 12, sbig = 12
34 > 12
↓~~

~~fbig = 34
sbig = 34
32 < 34~~

~~34, 12, 32, 54, 45, 6, 17
32 < 34
32 < 54
32 < 45
32 < 6
32 < 17
34 < 54
34 < 45
34 < 6
34 < 17
54 < 45
54 < 6
54 < 17
45 < 6
45 < 17
6 < 17
17 > 6~~

* Reverse the Array Elements Without Using Another Array!

class Main1

{

psvm()

{

int arr[] = {12, 34, 82, 54, 45, 6, 17};

int n = arr.length;

Sop("Number of Elements : " + n);

int temp;

for (int i=0; i < n; i++)

{ temp = arr[i];

arr[i] = arr[n-i-1];

arr[n-i-1] = temp;

} Sop("After Reversing");

for (i=0; i < n; i++)

{ Sop(arr[i]);

class main1

{ static void disp(

psvm()

{

int arr[] = new int[10];

for (int i=0; i < arr.length; i++)

arr[i] = 9 - i;

disp(arr);

arr[i] = arr[arr.length - 1 - i];

Sop(arr);

disp(arr);

Assignment

- * Sum & Avg of Array Elements
- ④ 1st least & 2nd least Element from the Array without sorting
- ⑤ WAP to find the sum of even positions & odd position elements

Sum & Avg of Array

```

import java.util.Scanner;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no of Elements");
        int n = sc.nextInt();
        System.out.println("Enter " + n + " Elements");
        int ar[] = new int[n];
        int sum = 0;
        for (int i = 0; i < ar.length; i++)
        {
            ar[i] = sc.nextInt();
            sum = sum + ar[i];
        }
        System.out.println("Sum of " + n + " Elements " + sum);
        System.out.println("Avg of " + n + " Elements " + (sum / ar.length));
    }
}

```

~~1st least & 2nd least Element from the Array without sorting~~

```
import java.util.Scanner;  
class main  
{  
    public static void main(String[] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the no. of Elements");  
        int n = sc.nextInt();  
        System.out.println("Enter " + n + " Elements");  
        int arr[] = new int[n];  
        int fl = arr[0], sl = arr[0];  
        for (int i = 1; i < n; i++)  
        {  
            if (arr[i] < arr[fl])  
            {  
                sl = fl; // (if i > fl) arr[i] will be sl  
                fl = arr[i];  
            }  
            else if (arr[i] < arr[sl])  
            {  
                sl = arr[i]; // (if i > fl) arr[i] will be sl  
            }  
        }  
        System.out.println("1st least element from Array " + fl);  
        System.out.println("2nd least Element from Array " + sl);  
    }  
}
```

Sum of Even & Odd position Elements in an Array

```
class main
{
    psum()
    {
        int arr[], n;
        Scanner sc = new Scanner(System.in);
        Sop("Enter no of Elements");
        int n = sc.nextInt();
        Sop("Enter the Elements of the Array");
        for (i=0; i<n; i++)
        {
            arr[i] = sc.nextInt();
        }
        int es=0, os=0;
        for (i=0; i<n; i++)
        {
            if (i%2==0)
                es = es + arr[i];
            else
                os = os + arr[i];
        }
        Sop("Sum of Even position Elements" + es);
        Sop("Sum of odd position Elements" + os);
    }
}
```

11/5/16 Q. SWAP the 1st part Array Element
with the 2nd part.

Ex:-

$$arr[] = \{2, 3, 4, 5, 6, 7, 8\}$$

(Initial state, so far, (1st part) formed (1) & 2nd part is formed (2nd part))

$$arr \Rightarrow 6 7 8 5 2 3 4$$

class main

{

psvm() { ("Swapping") for swap arr[0] with arr[1]

{

int arr[] = {2, 3, 4, 5, 6, 7, 8};

(Initial state, so far, 1st part is formed (1) & 2nd part is formed (2nd part))

int n = arr.length;

int temp; // for number

for (i=0; i < n/2; i++) { ("After") } 90%

{ temp = arr[i]; arr[i] = arr[n-3+i]; arr[n-3+i] = temp; }

arr[n-3+i] = temp; // after this arr[i] = arr[5]

arr[n-3+i] = temp; // after this arr[i] = arr[5]

SOP ("After Swapping");

for (i=0; i < n; i++)

SOP(arr[i]);

(i++ > i; i++))

(i++ > i; i++))

(i++ > i; i++))

* INSERT the Element into Existing Array in Specified Position

class main

{

static int int[] insert (int a[], int ele, int inx)

{ if (inx < 0 || inx >= a.length)

{ Sop ("index out of bound");

return a;

~~for~~: int na[] = new int[a.length + 1];

for (i=0; i < inx; i++)

na[i] = a[i];

na[inx] = ele; for (i=inx; i < a.length; i++)

na[i+1] = a[i];

PSVM()

{

return na; }

Sop ("Enter no. of elements");

int n = sc.nextInt();

int ar[] = new int[n];

Sop ("Enter Elements into Array");

for (int i=0; i < ar.length; i++)

ar[i] = sc.nextInt();

ar = insert(ar, 9, 3);

for (i=0; i <= n; i++)

Sop(ar[i]);

}

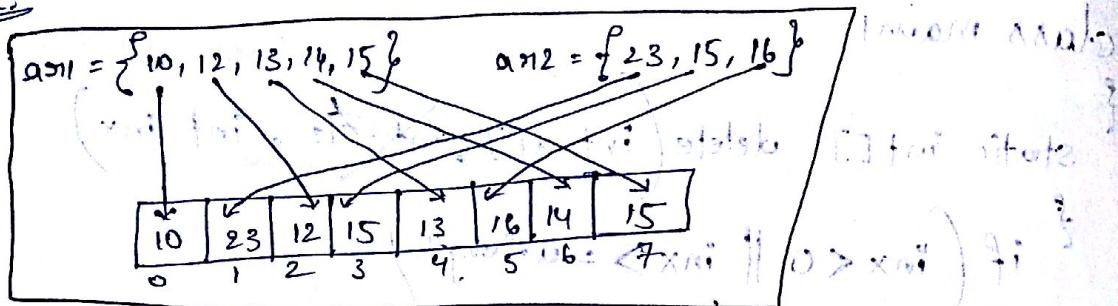
* DELETE the Elements from Existing Array in Specified position

```
class main1 {  
    static int[] delete(int a[], int i, int inx)  
    {  
        if (inx < 0 || inx >= a.length)  
        {  
            System.out.println("index out of bound");  
            return a;  
        }  
        int na[] = new int[a.length - 1];  
        for (int i=0; i < inx; i++)  
            na[i] = a[i];  
        for (int i=inx+1; i < a.length; i++)  
            na[i-1] = a[i];  
        return na;  
    }  
    public static void main(String args[])  
    {  
        System.out.println("Enter number of Element you want");  
        int n = sc.nextInt();  
        int ar[] = new int[n];  
        System.out.println("Enter Elements into Array");  
        for (int i=0; i < ar.length; i++)  
            ar[i] = sc.nextInt();  
        ar = delete(ar, 2);  
        for (int i=0; i < n-1; i++)  
            System.out.println(ar[i]);  
    }  
}
```

S/P
Enter number of Element you want
5
Enter Elements into Array
1 2 3 4 5
⇒ 1 2 4 5 → After deleting one index
from Array

* MERGE 2 Arrays in ZigZag manner

Ex:-



```

class main {
    public static void main(String[] args) {
        int arr1[] = {12, 34, 56, 78, 13};
        int arr2[] = {10, 7, 61, 2};

        display(arr1);
        SOP("-----");
        display(arr2);
        SOP("-----");

        int na[] = new int[arr1.length + arr2.length];
        int i, k;
        SOP("-----");

        for (i = 0, k = 0; i < arr1.length && i < arr2.length; i++) {
            na[k] = arr1[i];
            SOP(" " + arr1[i]);
            k++;
            SOP(" " + arr2[i]);
            SOP(" " + arr2[i]);
        }

        for (; i < arr1.length; i++, k++)
            na[k] = arr1[i];
        SOP(" " + arr1[i]);
        SOP(" " + arr1[i]);

        for (; i < arr2.length; i++, k++)
            na[k] = arr2[i];
        SOP(" " + arr2[i]);
        SOP(" " + arr2[i]);

        display(na);
        SOP("-----");
    }

    static void display(int a[])
    {
        for (int i = 0; i < a.length; i++)
            SOP(a[i] + " ");
        SOP("\n");
    }
}

```

* By Comparing 2 Arrays - find Distinct and Common Elements in the Array

```
class Main
```

```
{ static void display(int arr)
```

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

```
{ int arr1[] = {12, 34, 5, 67, 13};
```

```
int arr2[] = {34, 56, 78, 12};
```

```
display(arr1);
```

```
SOP("-----");
```

```
display(arr2);
```

```
SOP("Common Elements : ");
```

```
for (int i=0; i<arr1.length; i++)
```

```
{ for (int j=0; j<arr2.length; j++)
```

```
{ if (arr1[i] == arr2[j])
```

```
{ SOP(arr1[i] + " ");
```

```
break;
```

```
}
```

~~SOP~~
Sopln();

```
Sopln("Distinct Element");
```

```
for (int i=0; i<arr1.length; i++)
```

```
{ int find = 0;
```

```
for (int j=0; j<arr2.length; j++)
```

```
{
```

if ($\text{arr1}[i] == \text{arr2}[j]$)

{
 cout << "Elements matched" << endl;
 find = 1;

 break;

}
if (find == 0)

SOP ($\text{arr1}[i] + " "$);

for (int i=0 ; i< $\text{arr2}.\text{length}$; i++)

{
 int find = 0;

 for (int j=0 ; j< $\text{arr1}.\text{length}$; j++)

 if ($\text{arr2}[i] == \text{arr1}[j]$)

{

 find = 1;

 break;

}
if (find == 0)

SOP ($\text{arr2}[i] + " "$);

(ii) $\text{arr1} \rightarrow \text{arr2}$ if

• " $\text{arr1} \rightarrow \text{arr2}$ "

* Highest Contiguous Sum of 2 Elements from the Array

class main

{

psum() { }

int arr[] = { 10, 3, 2, 15, 13, 16, 9, 8 };

int sum = arr[0] + arr[1];

int k = 0;

for (int i = 1; i < arr.length - 1; i++)

{ if (sum < arr[i] + arr[i + 1])

arr[i + 1] = arr[i] + arr[i + 1];

sum = arr[i] + arr[i + 1];

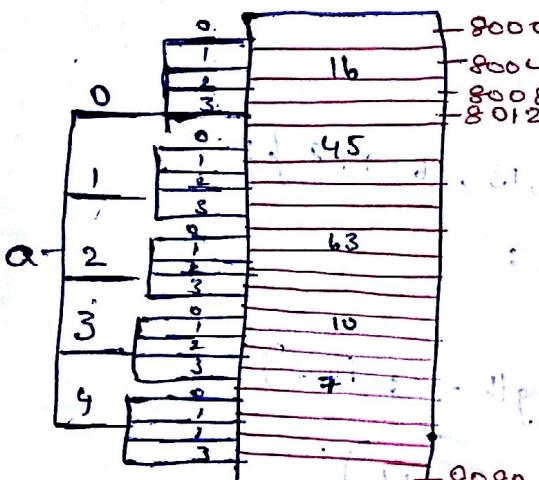
k = i;

SOP(sum);

SOP(k + " --->" + arr[k]);

SOP(k + 1 + " --->" + arr[k + 1]);

2-D Array [2 Dimensional Array]



int a[][];

a2 = new int[5][4];

No of Rows

No of Element

$$\rightarrow a[0][1] = 16$$

$$\rightarrow a[1][0] = 45$$

$$\rightarrow a[2][0] = 63$$

$$\rightarrow a[3][0] = 10$$

$$\rightarrow a[3][3] = 7$$

* a2.length \Rightarrow No of 1-D Rows

* $\rightarrow a2[0].length \Rightarrow$ No of elements in 1st Row

$\rightarrow a2[1].length \Rightarrow$ No of elements in 2nd Row

$\rightarrow a2[2].length \Rightarrow$ No of elements in 3rd Row

Ex:-

int a[][] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9, 10}, {1, 8} };

$\rightarrow a2.length$ 4

$\rightarrow a2[0].length$ 3

$\rightarrow a2[1].length$ 3

$\rightarrow a2[2].length$ 4

$\rightarrow a2[3].length$ 2

```
class main
```

```
{  
    psum( )  
    {  
        int arr[ ][ ] = {{ {17, 18, 20, 22}, {31, 32, 45} }, { {8, 6, 7, 4},  
            { {3, 5, 2, 1} } };  
        for (int i=0; i<arr.length; i++)  
            for (int j=0; j<arr[i].length; j++)  
                System.out.println("+" + arr[i][j] + "+");  
        System.out.println();  
    }  
}
```

④ Biggest Element in the matrix

Row wise & Column wise

```

class main {
    static void display (int mat[][]) {
        for (int i=0; i<mat.length; i++) {
            for (int j=0; j<mat[i].length; j++) {
                System.out.println(mat[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("Row sum");
        for (int i=0; i<mat.length; i++) {
            int rsum = 0;
            for (int j=0; j<mat[i].length; j++) {
                rsum += mat[i][j];
            }
            System.out.println("Row sum is " + rsum);
        }
        System.out.println("Column sum");
        for (int i=0; i<mat.length; i++) {
            int csum = 0;
            for (int j=0; j<mat[i].length; j++) {
                csum += mat[j][i];
            }
            System.out.println("Column sum is " + csum);
        }
    }
}

```

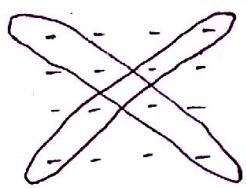
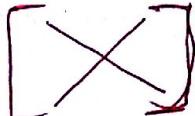
int arr[][] = {{17, 38, 20, 22}, {31, 32, 45, 17}, {18, 66, 27, 14}, {12, 34, 45, 32}};

display (arr);

for (int i=0; i<arr.length; i++) {
 int rbig = arr[i][0];
 int cbig = arr[0][i];
 for (int j=1; j<arr[i].length; j++) {
 if (rbig < arr[i][j]) {
 rbig = arr[i][j];
 }
 if (cbig < arr[j][i]) {
 cbig = arr[j][i];
 }
 }
 System.out.println("Row biggest element " + rbig);
 System.out.println("Column biggest element " + cbig);
}

(*) Biggest Element in the matrix

Diagonals



class main

{

psum()

{ int arr[][] = { {

display(arr);

int pbig = arr[0][0];

int sbig = arr[0][arr[0].length - 1];

for (int i=0; i<arr.length; i++)

{ for (int j=1; j<arr[i].length; j++)

{ if (i == j)

if (arr[i][j] > pbig)

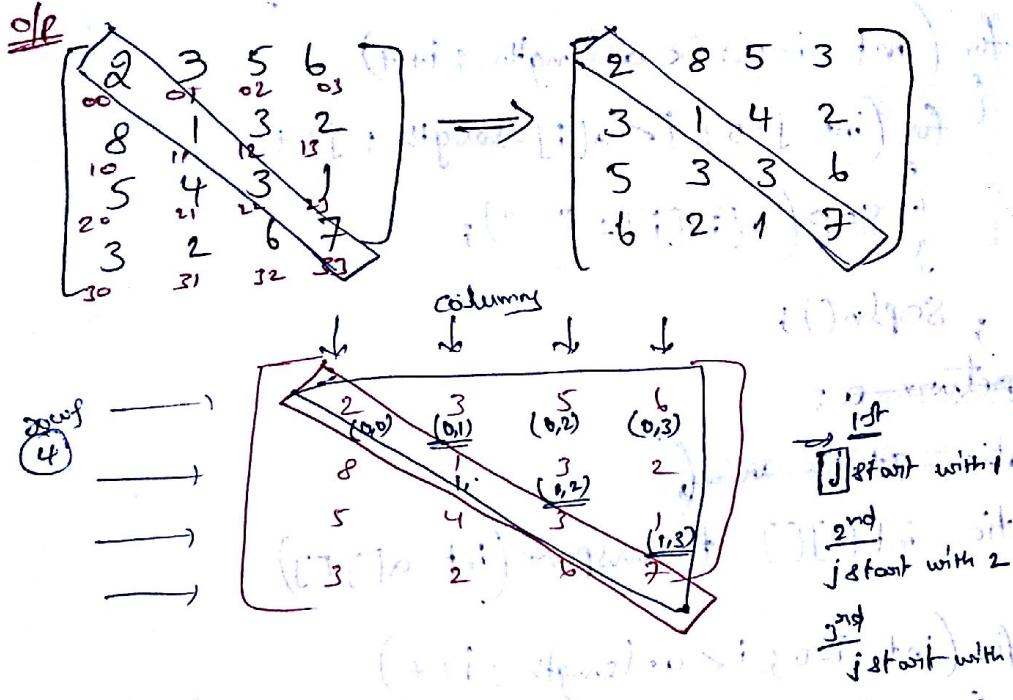
Pbig = arr[i][j];

if (~~i+j~~ == arr.length - 1)

if (arr[i][j] > sbig)

} } Sbig = arr[i][j];

~~POSE~~ (*) TRANSPOSE the Given matrix



```

import java.util.Scanner;
class Main
{
    public static void main()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the order of square matrix");
        int n = sc.nextInt();
        int mat[][] = new int[n][n];
        System.out.print("Enter " + n * n + " elements");
        for(int i=0; i<n; i++)
        {
            for(int j=0; j<n; j++)
            {
                mat[i][j] = sc.nextInt();
            }
        }
        System.out.print("Given matrix");
        display(mat);
        mat = transpose(mat);
        System.out.print("After transpose");
        display(mat);
    }
    static int[][] transpose(int mat[][])
    {
        int transposed[][] = new int[n][n];
        for(int i=0; i<n; i++)
        {
            for(int j=0; j<n; j++)
            {
                transposed[j][i] = mat[i][j];
            }
        }
        return transposed;
    }
}
  
```

Static void display(int a[][])

{

for (int i=0; i<a.length; i++)

{ for (int j=0; j<a[i].length; j++)

{ System.out.print(a[i][j] + " ");

} System.out.println();

} return a;

~~public static void~~

static int[][] transpose(int a[][])

{

for (int i=0; i<a.length; i++)

{ for (int j=i+1; j<a[i].length; j++)

{ int temp = a[i][j]

a[i][j] = a[j][i]

a[j][i] = temp;

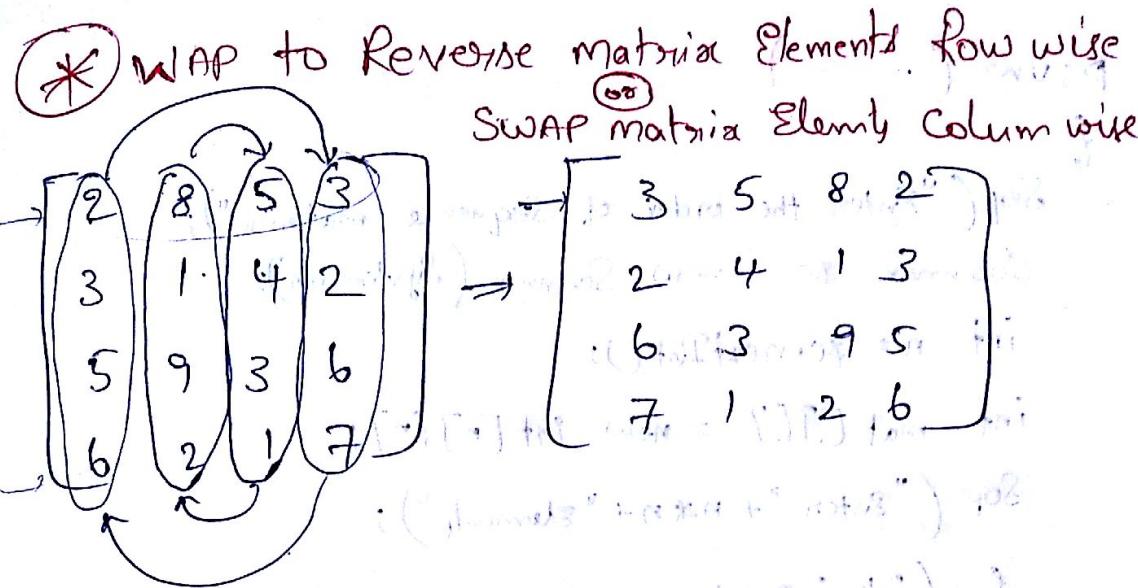
} } // transpose example do nothing with matrix

return a;

}

(int) [int] arr [int] [int] from file

if ("transpose" == str[1]) result = a



class main1

{

static void display (int a[][])

{

for (int i=0 ; i< a.length ; i++)

{ for (int j=0 ; j< a[i].length ; j++)

System.out.print(a[i][j] + " ");

System.out.println();

static int [][] rev (int a[][])

{

for (int i=0 ; i< a[i].length ; i++)

{ for (int j=0 ; j< a[i].length/2 ; i++)

{ if (a[i][j] != a[i][a[i].length - 1 - j])

{ int temp = a[i][j];

a[i][j] = a[i][a[i].length - 1 - j];

a[i][a[i].length - 1 - j] = temp;

g[i][j] = a[i][a[i].length - 1 - j];

g[i][j] = a[i][a[i].length - 1 - j];

return a;

PSUM (---)

{

```

Sop("Enter the order of Sequence matrix");
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
int mat[][] = new int[n][n];
Sop("Enter " + n * n + " elements");
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        mat[i][j] = sc.nextInt();
    }
}
Sop("Given matrix");
display(mat);
mat = rev(mat);
Sop("After rev matrix");
display(mat);
    
```

g

Row-Wise Exchange

1	2	3	4
5	6	7	8
9	7	6	8
6	5	4	2

→

6	5	4	2
9	7	6	8
5	6	7	8
1	2	3	4

Static int[] rev(int i[]){}

for (int i=0; i<a.length/2; i++)

if (i < a.length/2) for (j=0; j<a[i].length; j++)

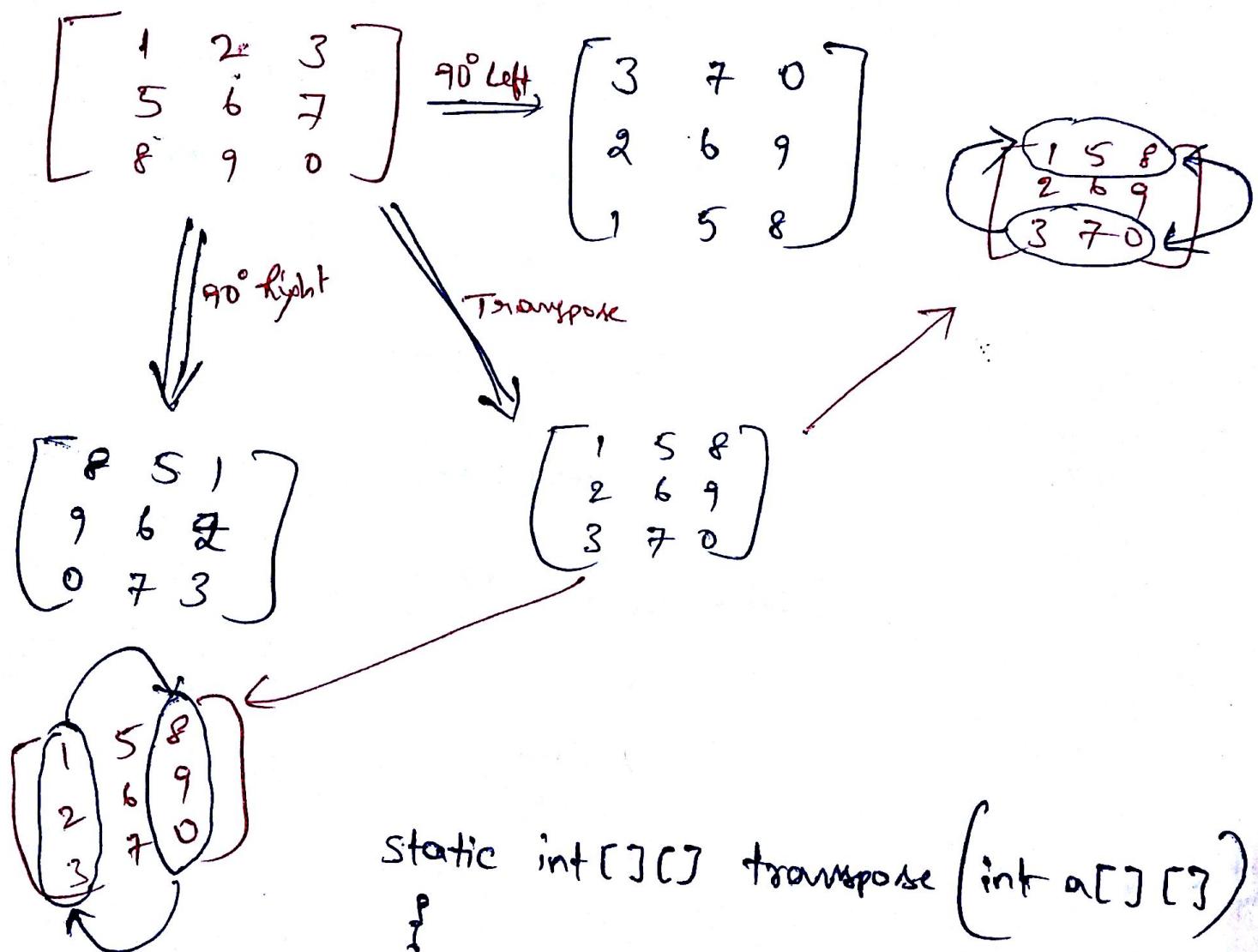
- int temp[0] = a[i][j]

a[i][j] = a[a.length-1-i][j]

a[a.length-1-i][j] = temp[0];

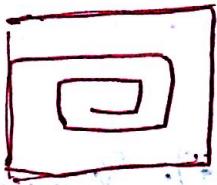
g

* Rotate the Given matrix into
 - 90° Right & 90° Left



17/5/16 * Display the matrix in Spiral order

2	3	4	5
6	7	8	9
8	3	4	7
9	8	7	5



class main

```
{ psum (-) : (int) sum = 0; int mat[4][4] = {{2,3,4,5},{6,7,8,9},{8,3,4,7},{9,8,7,5}};
```

```
for (int i=0, j=mat.length-1; i<j; i++, j--) { for (int k=i; k<j; k++) Sop (mat[i][j] + " "); for (int k=i; k<j; k++) Sop (mat[i][j] + " "); for (int k=j; k>i; k--) Sop (mat[i][k] + " "); for (int k=j; k>i; k--) Sop (mat[i][k] + " "); if (n % 2 != 0) Sopln (mat[mat.length/2][mat.length/2]); }
```

(*) Convert the Upper Case letter into Lower Case

Lower Case to Upper Case

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

```
class Main
```

```
{ public static void main()
```

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

```
    System.out.println("Enter your String");
```

```
    String st = sc.nextLine();
```

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

```
    char ch[] = st.toCharArray();
```

```
    for (int i = 0; i < ch.length; i++)
```

```
{
```

```
    if (ch[i] >= 65 && ch[i] <= 90)
```

```
        ch[i] = (char) (ch[i] + 32);
```

```
    else
```

```
        if (ch[i] >= 97 && ch[i] <= 122)
```

```
            ch[i] = (char) (ch[i] - 32);
```

```
    st = new String(ch);
```

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

nextLine → taking String

Value with Spaces Also

`javap java.lang.Object` \Rightarrow It gives skeleton of object class

`javap java.lang.String` \Rightarrow It gives entire methods present in Constructors which is present in String library.

* Given String is PALINDROME or not

With out ~~reversing~~ Reversing a String

Ex:

S	A	V	A	S
---	---	---	---	---

class main

{

psum (- - -)

{

SOp ("Enter your string");

String st = sc.nextLine();

SOp (st);

char ch [] = st.toCharArray();

int f=1;

for (int i=0; i < ch.length/2; i++)

{ if (ch[i] != ch[ch.length-i-1])

{

f=0;

break;

}

if (f==1)

SOpprintf ("%s is palindrome" + st);

else SOpprintf ("%s is not palindrome" + st);

18/5/16 • (*) Count the num of occurrences of each character in a given String

class main

{

psum (---);

SOP ("Enter your string");

Scanner sc = new Scanner (System.in);

String st = sc.nextLine();

char ch[] = st.toCharArray();

int n = ch.length;

for (int i=0; i<n; i++)

{

int count = 1;

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

{

if (ch[i] == ch[j])

{

for (int k=j+1; k<n; k++)

{

ch[k] = ch[k+1];

(n-j) + 1 - 1 = (n-j)

j--;

}

SOP (ch[i] + " --> " + count);

ababcdeffgabc
 a → 3
 b → 3
 c → 2
 d → 1
 e → 1
 f → 1
 g → 1

(*) Remove the Repeated character from
String Except Space irrespective of cases

```
class main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String st = sc.nextLine();
        char ch[] = st.toCharArray();
        int n = ch.length;
        st = "";
        for (int i = 0; i < n; i++)
        {
            st = st + ch[i];
            for (int j = i + 1; j < n; j++)
            {
                if ((ch[i] != ' ') && (ch[i] == ch[j] || ch[i] == ch[j] + 32 || ch[i] == ch[j] - 32))
                {
                    ch[j] = ch[j + 1];
                    n--;
                    j--;
                }
            }
        }
        System.out.println(st);
    }
}
```

class A {
 final int u1;
 final int u2;
 A (int u1, int u2)
 {
 this.u1 = u1;
 this.u2 = u2;
 }
 }
 to make immutable

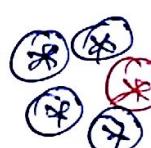
$$\begin{array}{l} u1 = 10 \\ u2 = 20 \end{array}$$

a1

class A {
 final int u1;
 final int u2;
 A (int u1, int u2)
 {
 this.u1 = u1;
 this.u2 = u2;
 }
 }
 a1 = new A(10, 20);
 a1.u1 = 45; (X)
 a1.u2 = 200;

a1 = new A(50, 40);

a1 → $\begin{array}{l} u1 = 50 \\ u2 = 40 \end{array}$



count

No. of words in a given String

class Main {
 {
 PSVM();
 }
}

32 is ASCII value of Space

```

PSVM()
{
    S0P("Enter your String");
    Scanner sc = new Scanner(System.in);
    String st = sc.nextLine();
    char ch[] = st.toCharArray();
    int n = ch.length;
    int count = 1;
    for(int i=0; i<n-1; i++)
    {
        if((ch[i] == ' ' && ch[i+1] != ' ') || ch[i] == ',' && ch[i+1] != ',')
            count++;
    }
}

```

$\| \text{ch}[i] == '.' \&\& \text{ch}[i+1] == '!' \&\& \text{ch}[i+1] != ' ' \&\& \text{ch}[i+2] == '!$

Count ++;

{
} statements
{
} statements
{
} statements
{
} statements

{
} statements
{
} statements
{
} statements

* Count of no. of Vowels, Consonants, numbers and Special characters.

no. of vowels = count1

no. of consonants = count2

no. of numbers = count3

no. of special characters = count4

for (int i=0; i<n; i++)

{ if ($\text{ch}[i] == 'a' \|| \text{ch}[i] == 'A' \|| \text{ch}[i] == 'e' \|| \text{ch}[i] == 'E' \|| \text{ch}[i] == 'i' \|| \text{ch}[i] == 'I' \|| \text{ch}[i] == 'o' \|| \text{ch}[i] == 'O' \|| \text{ch}[i] == 'u' \|| \text{ch}[i] == 'U'$)

Count1++;

else if ($(\text{ch}[i] >= 'a' \&\& \text{ch}[i] <= 'z') \|| (\text{ch}[i] >= 'A' \&\& \text{ch}[i] <= 'Z')$)

Count2++;

else if ($\text{ch}[i] >= '0' \&\& \text{ch}[i] <= '9'$)

Count3++;

else {

Count4++;

Sop("Vowels --> " + count1); Sop("numbers --> " + count3);

Sop("Consonants --> " + count2); Sop("Special characters --> " + count4);

19/5/11

Inner Class

Class containing

class A

{

int a=10

static int b=20;

{

}

Static void f1()

{

}

Static

{

}

{

}

class B

{

=

{

static class B

{

}

} DataMember

} methods

} Blocks

} Subclasses

Inner class

Whenever class prefixed by static

then we can call that class is
Inner class or Nested class.

* A class inside Another class.

Types

1. STATIC inner class
2. Non-Static inner class
3. Local inner class
4. Anonymous inner class

* What are the different types of members we have in class?

① Static

- Variable

- method

- Block

- Subclass (Inner class)

② Non-Static

- Variable

- method

- Block

- Subclass (Inner class)

STATIC inner class

```

class Sample {
    static int a = 30;
    static int b = 40;
}

static class Demo {
    static int a1 = 40;
    static int b1 = 50;
}

void m() {
    static int a2 = 50;
}

```

{ Static members of outer class can be accessed by object of outer class}

Sop(Sample.b) Sop(Demo.b)

Sopn(Sample.b, Demo.b) Sopn(new Sample().a)

Sample.Demo sd = new Sample.Demo();
Sop(sd.a1);

NOTE :-

- ① Through inner Class we can access private members of outer class members
- ② If we want to access only data members of inner class, No need to create instance of outer class

Static Inner Class

- ① If you define a class inside another class prefixed by a static keyword is called "Static inner class"
- ② Static inner class can have
 - (i) Static members
 - (ii) Non-Static members
- ③ Static members of static inner class can be accessed without creating object of outer class as well

④ Inside static inner class, we can access outer class static data members including private but not possible to access instance members of outer class

⑤ Static members of nested class we can possible to Access outerclass.name • innerclass.name • static member

⑥ If you want to Access instance member of static inner class no need to create a object for outer class

public class Sample

{ int a=10;

static int b=20;

static class Demo

{ int a1=40; static int b1=50;

void m1()

{ System.out.println("i am m1 of sample"); }

SOP(b);

SOP(new Sample().a);

SOP("i am m1 of demo");

static void m2()

SOP("i am m2 of demo");

void m3()

SOP(Demo.b); SOP(new Demo().a);

- ① Inside a Instance inner class, we are not possible to define the static members; only Non-Static members
- ② We can Access static mem's of outer class with out creating instance of outer class
- ③ We can Access static & Non-static members of outer class directly with out creating instance of outer class
- ④ If you want to Access instance members of instance inner class in the outer class, we have to create instance of inner class
- ⑤ If you want to Access ~~instance~~ inner class members of instance inner class, we have to Create instance of inner class as well as outer class.

Non-Static Inner Class

Public class Sample

{
 int a=10;

 static int b=20;

 class Demo

 {
 int a1=40;

 static int b1=50; X

 Void m1()

 SOP(b);

 SOP(a);

 SOP("i'm m1 of Demo");

 Static Void m2()

 SOP("i'm m2 of Demo");

 Void msg()

 SOP(new Demo().a1);

~~new Demo~~

 SOP(new Demo().m1());

 }

 }

 class main

{

 PSVM(- - -)

 {
 SOP("-----");

 Sample s1 = new Sample(); // outer class object

 Sample.Demo d1 = s1.new Demo(); // inner class object

 SOP(d1.a1); d1.m1();

Instance inner
class

20/5/16

LOCAL INNER CLASS

We can Create in
methods
Blocks

```
class Sample {  
    void m1() {  
        static class A {  
            private int a = 20; // ✗  
            int a; static int b; // ✗  
            void m2() {  
                {  
                    A a1 = new A(); // ✗  
                    System.out.println(a1.a); // ✗  
                }  
            }  
        }  
    }  
}
```

In local class,
we can't create

A a1 = new A(); // ✗
we can't create
instance of local class

in outside method

- ① Define a class, Inside a method or blocks that type of class is called
- ② If you want to invoke method of local inner class
- ③ If you want to Access the member of local inner class we have to instantiate the local inner class inside that method only
- ④ Local inner class can't be invoked from outside the method.
- ⑤ Inside a local inner class, we can Access outer class instance members
- ⑥ for local variables of local inner class, we can't use Access Specifiers

⑦ Non-final local variables are not possible to access inside the local inner-class

Anonymous class local inner class

Public class Main

```
{  
    public static void main (String args[]) {  
        SOP ("-----");
```

```
        Student st1 = new Student (123, "Rakesh", 73.28);  
        SOP (st1);
```

```
        SOP (st1.hashCode());
```

```
        Student st2 = new Student (124, "Suresh", 53.83);  
    }
```

```
    public String toString () {
```

```
        return "hi swush";
```

```
    public int hashCode () {
```

```
        return id;
```

```
    void mi () {
```

```
        SOP ("I am mi of Anonymous");
```

```
    }  
};
```

object.mi();

package com.jspidery.jave

public class Student

{ int id;

String name;

double per;

public Student (int id, String name, double per)

{
super();

this.id = id;

this.name = name;

this.per = per;

public String toString()

return "Student [" + name + "]";

Blocks

Types

- ① SIB (Static Initializer Block)
- ② IIB (Instance Initializer Block)

static
{}
}

{
}
}

IIB

class A
{
{

IIB

Instance Initializer Block

System.out.println("I am IIB of class A");
}

A (int a)

{ super(); → implicitly
{
}

A (int a, int b)

{ super(); → implicitly
{
}

A ()

{ super(); → implicitly
{
}

class Main {

{ Psum (...) {
{

 A a1 = new A();

 A a2 = new A(10);

 A a3 = new A(10, 20);

}
}

Output

I am IIB of class A

I am SIB of class A

I am SIB of class A

* After the compilation, IIB gets copied after the Super calling stmt in Every Constructor.

* If you execute any task immediately after creating an instance of that class, use IIB in our code.

① IIB is used for initialize the default values of ^{non-static} data members of that class

* Here we get 0p

0 ~~0~~ default IIB gives

default values to the
Data mem's of class

```
class A
{
    int a;
    int b;
}
SOP(a); 0
SOP(b); 0
```

```
class main()
{
    public static void main()
    {
        A a = new A();
    }
}
```

② IIB is used for initialize the Instance final variable

* We can Initialize the Instance final variable Either

① non-static-Block (IIB)

② Constructor

③ Declaration time

final int k = 10;

```
①
public class psample
{
    {
        SOP("Welcome to psample");
    }
    psample()
    {
        SOP("i am psample cons... ");
    }
}
```

```
③
public class Main
{
    {
        SOP("i m iib of main");
    }
    psum (----)
    {
        SOP("mms");
        Sample s1 = new Sample();
    }
}
```

```
②
class Sample Extends Sample
{
    int a,b;
    final int f;
    double pi;
    {
        f = 900;
        a = 20;
        b = 90;
        pi = 3.143;
        SOP(" Welcome to Sample");
    }
    Sample()
    {
        SOP(" i am no arg cons of Samp");
        Sample(int a)
        {
            SOP("i am 1 arg cons of Samp");
            Sample(int a,int b)
            {
                SOP("i am 2 arg cons of Samp");
            }
        }
    }
}
```

SOP("-----");

Sample $\&_2$ = new Sample(10);

SOP("-----");

Sample $\&_3$ = new Sample(10, 20);

}

g

① class parent

{ static

{ SOP("i am sib of parent");

{ SOP("i am iib of parent");

parent()

{ SOP("i am cons of parent");

③ class Main2

{ static

{ SOP("i am iib of Main2");

psum

{"-----");

{ "i am sib of mms";

SOP("mms");

child c1 = new child(10, 20);

SOP("-----");

child c2 = new child(10);

SOP("-----");

child c3 = new child();

SOP("-----");

dp

mms

i am Psample copy

psum(10, 20);

child c1;

class child Extends parent

{

static

{ SOP("i am sib of child");

{ SOP("i am iib of child");

child();

{

SOP("i am cons of child");

child (int a)

{ this();

SOP("i am larg. cons of child");

child (int a, int b)

{

SOP("i am & larg. cons of child");

child c1 = new child(10, 20);

SOP("-----");

child c2 = new child(10);

SOP("-----");

child c3 = new child();

SOP("-----");

main2 m2 = new main2();

dp

~~MAC~~

I am sib of main 2

MMS

I am sib of parent

I am sib of ~~parent~~ child

I am ~~sib~~ of parent

STATIC BLOCKS (SIB)

① SIB is used for initialize the default values of static data members of that class

② Inle Com. initialize static ^{final} Variable Either

① Declaration time

② Static - Blocks (SIB)

③ If you execute Any task only one time when creating an instance of Specific class. throughout Application, use SIB in your class

① To find the middle element from the given String:

→ abcdef

② find the sum of odd elements from the Array.

③ ~~Replace~~ Replace vowels with * in given string.

④ Replace the sequence character by * Raaghhhhi

⑤ WAP to search element from the given Array if found reverse that number:

10 532 384 51

$$A = \{10, 235, 384, 15\}$$

$$B = \{235, 16, 75, 15\}$$

compare

⑤ WAP to Replace the Special character by * in a given string.

① MIDDLE Element

```
class main
{
    p8vm (---)
    {
        char[] ch = { 'a', 'b', 'c', 'd', 'e' };
        String str = "abcde";
        char[] ch = str.toCharArray();
        for (int i = 0; i < ch.length; i++)
        {
            if (ch.length / 2 == 0)
            {
                SOP (ch[ch.length / 2] + ch[ch.length / 2]);
            }
            else
                SOP (ch[ch.length / 2]);
        }
    }
}
```

③ Replace vowels with *

```
String str = "YESWANTH";
char[] ch = str.toCharArray();
String res = "";
for (int i = 0; i < ch.length; i++)
{
    if ((ch[i] == 'a' || ch[i] == 'A') ||
        ('e' || 'E' || 'i' || 'I' || 'o' || 'O'))
        SOP ("*");
    else
        res += ch[i];
}
SOP (res);
```

② Sum of odd Elements

```
class main
{
    p8vm (---)
    {
        char[] ch = { '1', '2', '3', '4', '5', '6', '7', '8' };
        int sum = 0;
        for (int i = 0; i < ch.length; i++)
        {
            if (ch[i] % 2 != 0)
                sum = sum + ch[i];
        }
        SOP ("Sum of odd Elms: " + sum);
    }
}
```

④ Replace Sequence character
by *

String str = "YEEESWAANthh"

char[] ch = str.toCharArray();

String res = "";

for (int i=0; i<ch.length; i++)

{ int flag=0;

while (ch[i] == ch[i+1])

{ flag=1; i++; }

}

if (flag==1)

{ res = res + "*" ; }

}

res = res + ch[i];

} res = res + ch[i];

⑤ Search Element from
given Array if found
reverse that num.

for (i=0; i<arr.length; i++) { for (i=0; i<arr.length; i++) {

{ for (j=0; j<arr.length; j++) { for (j=0; j<arr.length; j++) {

{ if (arr[i] == arr[j]) { if (arr[i] == arr[j]) {

{ while (arr[j]>0) { while (arr[j]>0) {

{ int temp = arr[j]/10; { int temp = arr[j]/10; {

rev = rev * 10 + temp; { rev = rev * 10 + temp; {

{ arr[j] = arr[j]/10; { arr[j] = arr[j]/10; {

{ sop (rev); { sop (rev); {

{ else { else {

{ sop (arr[i]); { sop (arr[i]); {

} } }

25/5/16 (*) Display the words in init cap manner in the String

O/P \Rightarrow chitradurga is a float city.

O/P \Rightarrow Chitradurga - Is A Font City.
 $i=0;$ $ch[i-1];$

Class main

{

PSUM (-----)

{

Scanner sc = new Scanner(System.in);

SOP(" Enter Sentence: ");

String str = sc.nextLine();

char[] ch = str.toCharArray();

String res = " ";

for (int i=0; i < ch.length; i++)

{

if ($i == 0 \text{ || } ch[i-1] == ' '$)

{

if ($ch[i] >= 97 \text{ && } ch[i] <= 122$)

res = res + (char)(ch[i] - 32);

else

res = res + ch[i];

else if ($ch[i] >= 65 \text{ && } ch[i] <= 90$)

res = res + (char)(ch[i] + 32);

else

res = res + ch[i];

SOP(res);

}

}

}

Reverse Words in a Sentence

IP chitradurga is a fort city.
OP agrudantih si a trof ytic

class Main

{
 public (- - -)

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

SOP ("Enter Sentence");

String str = sc.nextLine();

char[] ch = str.toCharArray();

String res = "";

for (int i=0; i<ch.length; i++)

{
 int k=i;

 while (i<ch.length & ch[i]!=' ')

{
 i++;

 int j=i-1;

 while (j>=k)

{
 res = res + ch[j];

 j--;

 res = res + " ";

}
SOP(res);

* Reverse Sentence

IP chitradurga is a fort, City

OP city fort, a is chitradurga

class Main

{

PSUM (---)

:{

Stop ("Enter Sentence: ");

String str = sc1.nextLine();

char[] ch = str.toCharArray();

String res = "";

Stop ("After Reversing the Sentence");

for (i = ch.length - 1; i >= 0; i - -)

{

int k = i; if (ch[i] >= 'A' & ch[i] <= 'Z')

while (i >= 0 && ch[i] != ' ')

{ i - -;

int j = i + 1;

while (j <= k)

{

res = res + ch[j];

j++;

res = res + " "

Sop (res);

}

* Reverse the Specified Word from the String

If chitradurga is a fort city

Enter word to be search \Rightarrow fort

or

fort is present

After reversing \Rightarrow chitradurga is (not) city

class main

{ psum (- - -)

? SOP ("Enter Sentence"); | SOP ("Enter word to be search");

String str = sc.nextLine(); | String str1 = sc.nextLine();

char[] ch = str.toCharArray(); | char[] ch1 = str1.toCharArray();

String res = "";

for (int k = 0, L = 0; count = 0;

for (int i = 0; i < ch.length; i++)

{ int j = 0;

while (i < ch.length && j < ch.length && ch[i] == ch[j])

{

if (count == 0) {

K = i;

res += ch[i];

i++;

j++;

{

L = j - 1;

if (j == ch.length && ch[i] == ' ') {

{ SOP (str1 + "is present");

SOP ("After reversing");

while (L >= K) {

{ res = res + ch[L];

} L--;

$\text{res} = \text{res} + \text{ch}[i];$

$\{ \}$
SOP(res);
 $\{ \}$

* Given two strings are

Ex:-

Anagram or not

- ① Keep {
- ② Peek } \Rightarrow these two strings are Anagram

Ex:-

- ① Keep {
- ② Peek } \Rightarrow NOT Anagram

Step 1 :- Removal of Spaces

Step 2 :- No. of characters

Step 3 :- Both the strings set to one case

Step 4 :- Sort 2 strings

Step 5 :- Compare

- ① Keep listen
- ② peek silent $\xrightarrow{①}$ keeplisten
peeksilent $\xrightarrow{②}$ $\begin{matrix} 10 \\ 10 \end{matrix}$ $\xrightarrow{③}$ keeplisten
peeksilent

Anagram \longleftrightarrow $\begin{matrix} e & e & k & l & n & p & s \\ e & e & i & k & l & n & p & s \end{matrix}$ $\xrightarrow{⑤}$

\downarrow $\xrightarrow{④}$
eeeiklnpst
eeeiklnpst

```

import java.util.Scanner;
class Anagram
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter string 1");
        String st1 = sc.nextLine();
        System.out.println("Enter string 2");
        String st2 = sc.nextLine();
        st1 = removeSpace(st1);
        st2 = removeSpace(st2);
        if(st1.length() != st2.length())
        {
            System.out.println("Strings are not an Anagram");
        }
        else
        {
            st1 = toLower(st1);
            st2 = toLower(st2);
            st1 = sort(st1);
            st2 = sort(st2);
        }
    }
}

```

method①

(a)

method②

if (st1.equals(st2))

System.out.println("Strings are an Anagram");

else

System.out.println("Strings are not an Anagram");

Printable
method

user defined
method

boolean eq = compare(st1, st2);

if (eq == true)

System.out.println("Strings are an Anagram");

else

System.out.println("Strings are not an Anagram");

static String removespace (String str)

{

String res = "";

char ch[] = st.toCharArray();

for (int i=0; i<ch.length(); i++)

{ if (ch[i] != ' ')

; res = res + ch[i];

; return res;

static String tolower (String st)

{ String res = "";

char ch[] = st.toCharArray();

for (int i=0; i<ch.length; i++)

{ if (ch[i] >= 65 && ch[i] <= 90)

res = res + (char)(ch[i]+32);

else

; res = res + ch[i];

; return res;

static String sort (String st) ~~String str~~

{

char ch[] = st.toCharArray();

for (int i=0; i<ch.length-1; i++)

{ ~~for (int j=i+1; j<ch.length; for (int j=i+1; j<ch.length;~~

if (ch[i]>ch[j]) { j++ }

char temp = ch[i]

if (ch[i] == ch[j])

ch[i] = temp;

; return new String(ch);

```

static boolean compare(String st1, String st2) {
    char ch1[] = st1.toCharArray();
    char ch2[] = st2.toCharArray();

    for (int i = 0; i < ch1.length; i++) {
        if (ch1[i] != ch2[i]) {
            return false;
        }
    }
    return true;
}

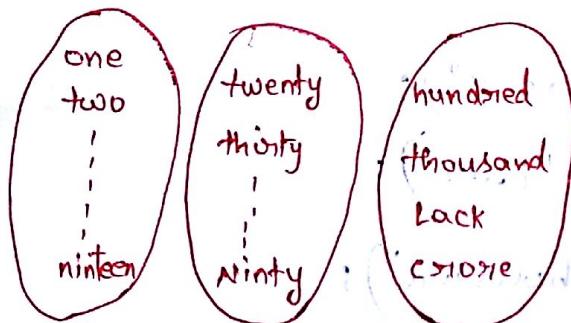
```

~~getSum~~

Display the Number in the form of Sentence.

1/p → 5927

1/p → Five thousand Nine hundred twenty Seven



These are all unique words.

class Main

```

{
    String one[] = {"", "One", "Two", "Three", "Four", "Five",
                    "Six", "Seven", "Eight", "Nine", "Ten", "Eleven",
                    "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen",
                    "Seventeen", "Eighteen", "Nineteen"};
}

```

String two[] = { "One", "Two", "Three", "Four", "Five",
"Six", "Seven", "Eight", "Nine", "Ten", "Eleven",
"Twelve", "Thirteen", "Fourteen", "Fifteen",
"Sixteen", "Seventeen", "Eighteen", "Nineteen",
"Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy",
"Eighty", "Ninety" };

```
static void pw (int n, String st) {
    if (n > 19)
        SOP (two[n/10] + one[n%10]);
    else
        SOP (one[n]);
    if (n != 0)
        SOP (st + " ");
}
```

Psum (- - - - -)

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

```
SOP ("Enter the Number");
```

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

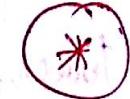
```
pw (n/10000000, "crore");
```

```
pw ((n/100000)%100, "Lakh");
```

```
pw ((n/1000)%100, "Thousand");
```

```
pw ((n/100)%10, "Hundred");
```

```
pw (n%100, "");
```



Count the Number of days between the Dates

~~Ex~~

28 - 2 - 2015 to 3 - 6 - 2016

start each iteration

?

0 0 0 0 0

+ MMYY + mm + dd + hh

m1

28 - 02 - 2015

difference
mm - mm, dd - dd

n2

3 - 06 - 2016

dd - dd, hh - hh

MMYY = MMYY + MMYY

MMYY = MMYY + MMYY

① if month <= 2

$$\left(\frac{2014}{4} - \frac{2014}{100} + \frac{2014}{400} \right) + 28 + m[0] + 2015 * 365$$

No. of Leap year current month previous month no. of Days
Days completed

② if month > 2

$$\left(\frac{2015}{4} - \frac{2015}{100} + \frac{2015}{400} \right) + \text{current month days} + \text{previous month days} + 2015 * 365$$

Days completed

Leap year. \Rightarrow

number / 4 & / 400 not
/ 100

num / 4 & num / 400 num! / 100

(Opinion of unwise writing)

" " + 6666 + " " + month + " " + day + " " + month

* Count the Number of Days Between the DATES

```
public class Date
```

```
{
```

```
    int dd, mm, yyyy;
```

```
    final int m[] = {31, 28, 31, 30, 31, 30, 31, 31, 30,
```

```
        31, 30, 31};
```

```
    public Date (int dd, int mm, int yyyy);
```

```
{
```

```
    this.dd = dd;
```

```
    this.mm = mm;
```

```
    this.yyyy = yyyy;
```

```
    public int numberOfDays ()
```

```
{
```

```
    int days = yyyy * 365;
```

```
    for (int i=0; i<mm-1; i++)
```

```
{
```

```
    days += m[i];
```

```
}
```

```
    if (mm > 2)
```

```
        days += (yyyy)/4 - (yyyy)/100 + (yyyy)/400;
```

```
    else
```

```
        days += (yyyy-1)/4 - (yyyy-1)/100 + (yyyy-1)/400;
```

```
    return days + dd;
```

```
}
```

```
.. public String toString ()
```

```
{
```

```
    return " " + dd + " ." + mm + " ." + yyyy + " ";
```

```
}
```

```
public class main2
```

```
{
```

```
    static Date readDate()
```

```
{
```

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

```
    System.out.println("Enter the dd");
```

```
    int dd = sc.nextInt();
```

```
    System.out.println("Enter the mm");
```

```
    int mm = sc.nextInt();
```

```
    System.out.println("Enter the yyyy");
```

```
    int yyyy = sc.nextInt();
```

```
    return new Date(dd, mm, yyyy)
```

```
}
```

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

```
{
```

```
    Date d1 = readDate();
```

```
    Date d2 = readDate();
```

```
    int count1 = d1.getDays();
```

```
    int count2 = d2.getDays();
```

```
    if (count1 > count2)
```

```
{
```

```
        System.out.println("Number of days b/n " + d1 + " And " +  
                           d2 + ":" + (count1 - count2));
```

```
    } else
```

```
{
```

```
        System.out.println("No of Days b/n " + d1 + " And " + d2 +  
                           ":" + (count2 - count1));
```

```
}
```

```
}
```



Sub-String

```
import java.util.Scanner; // importing Scanner class

class Main {
    static boolean check(String str1, String st) {
        char ch1[] = str1.toCharArray();
        char ch2[] = st.toCharArray();
        int count = 0;
        for (int i = 0; i < ch1.length; i++) {
            int j = 0;
            int k = i;
            while (j < ch2.length && i < ch1.length &&
                   (ch1[i] == ch2[j] || ch1[i] == ch2[j] + 32 ||
                    ch1[i] == ch2[j] - 32)) {
                if (ch1[i] == ch2[j] - 32) {
                    i++;
                }
                j++;
            }
            if (j == ch2.length) {
                count++;
            }
        }
        System.out.println("Count: " + count);
        if (count > 0) {
            System.out.println(st + " occurred " + count + " times");
            return true;
        }
    }
}
```

```

        return false;
    }

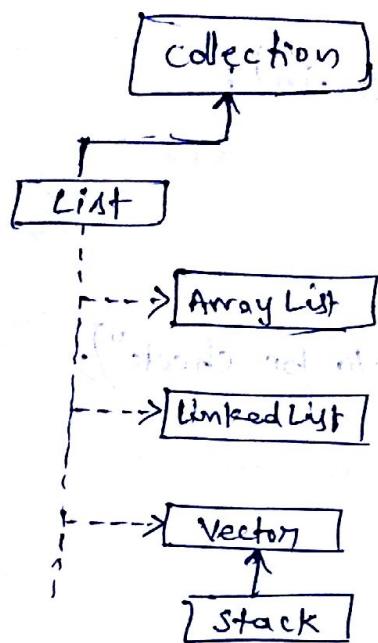
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter main String");
        String str = sc.nextLine();
        System.out.println ("Enter Substring (which one to be check)");
        String st = sc.nextLine();
        boolean res = check (str, st);

        if (res)
            System.out.println ("st + " + st + " present in the " + str);
        else
            System.out.println ("st + " + st + " not present in the " + str);
    }
}

i/p Enter string
How are you how are you doing?
Enter Substring
How | Your
1 time pos : 3 | your not present in the
2 time pos : 15 | string
How occurred 2 times
How present in the string.

```

* Implement your own ArrayList
Given by a name : Raghulist



* What are the prop of List?

- ① duplicates are allowed
- ② more than one null insertions allowed
- ③ heterogeneous
- ④ Insertion order maintained
- ⑤ preserved

ArrayList :- growable (3) Reliable Array

deletion ① insert an element from array [By using left and right shifts]

When it is used ? To Retrieve.

When it is not used ?

Insert & Deletion

Linked List :- doubly linked list

Data Structure :- organizing the memory for data is called Data Structure

When it is used ?

for Insertion & Deletion

When it is not used :-

When every frequent operation is Retrieval

* ArrayList having Overload Constructors

[javap: java.util.ArrayList] → This will give entire info about ArrayList class

types of Constructors

ArrayList arr = new ArrayList(); → Defaultly creates array with 10 elements

ArrayList arr = new ArrayList(8); → It creates array with size 8

ArrayList arr = new ArrayList(arr); → With same elements

(P) do nothing storing

(P) do nothing storing

(P) do nothing storing

collection c = new ArrayList();

* Here we are Casting ArrayList to collection.

So we can use only properties of collection interface.

add

addAll

removeAll

retainAll

clear

isEmpty

size()

toArray()

iterator()

* Implement your own ArrayList

given by a name Raghulist

(3)

JList

package first;

import java.util.collections;

public class JList : (1) implements Comparable

{

private Object ob[];

private int i=0;

private int cpy;

public JList()

{

cpy = 10;

ob = new Object[cpy];

public JList(int cap)

{

cpy = cap;

ob = new Object[cpy];

}

public boolean add(Object obj)

{

if (i < ob.length)

{

ob[i] = obj;

i++;

return true;

else

{

cpy = cpy * 3 / 2 + 1;

Object nob[] = new Object[cpy];

```

for (int i=0; i< ob.length; i++)
{
    nob[i] = ob[i];
}
nob[i] = obj; (([i] do something) if i > ob.length)
ob = nob;
i++;
return true; (i > ob.length)
}

public boolean addAll(Collection c)
{
    Object obj[] = c.toArray();
    for (int i=0; i< obj.length; i++)
    {
        add(obj[i]);
    }
    return true;
}

public int size()
{
    return i;
}

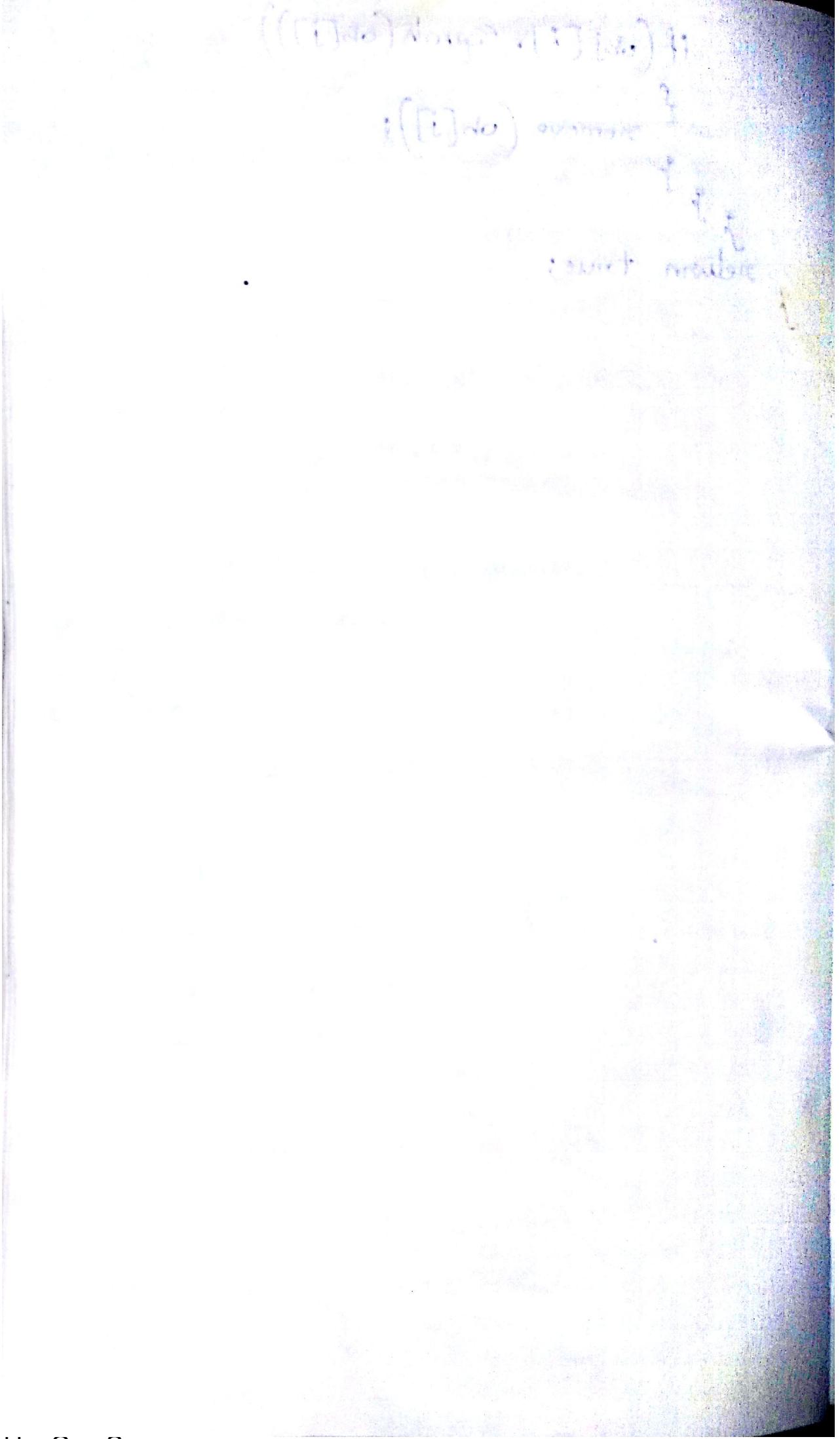
public int capacity()
{
    return cpy;
}

public boolean isEmpty()
{
    if (i==0) {return true;}
    else {return false;}
}

public void clear()
{
    ob = new Object[cpy];
    i=0; → for start index from 0
}

```

```
if (obj[i].equals(obj[j]))  
{  
    remove (obj[j]);  
}  
}  
return true;  
}
```



protection. We experienced some difficulty in getting air

Flowers grow well here.

(Great white pelican) Histored 17 minutes later

3) permutation: $\{ \} \rightarrow \{ \}$

$$f(x) = \frac{1}{2}x^2 + 2$$

(+/- sign depends on $i < 0 = i - \text{fix}$) \Rightarrow

$$(z_1 + \sqrt{d})^k$$

John Lang

(continued) $\frac{1}{2}$ mole of H_2 reacts with Cl_2 to produce HCl .

$\omega = 2$

(2-1134) 17

$$[1]_R + [2]_R = [3]_R$$

2000-2001

to do ~~and~~ ~~not~~

(part 17 of 28) - more like white salinity

* Split method [User-Defined].

the words and Interchange odd positions in String

```
public class main {
{
    static String[] breakIt(String str, char c) {
        char ch[] = str.toCharArray();
        int count = 0;
        for (int i=0; i<ch.length; i++) {
            if (ch[i] == c)
                count++;
        }
        String st[] = new String[count+1];
        int k=0;
        st[k] = "";
        for (int i=0; i<ch.length; i++) {
            if (ch[i] != c)
                st[k] = st[k]+ch[i];
            else
                k++;
            st[k] = "";
        }
        return st;
    }
    public static void main (String[] args) {
        Scanner sc1 = new Scanner (System.in);
    }
}
```

```
SOP("Enter your String");
```

```
String st = sc.nextLine();
```

```
String s[] = breakIt(st, ' ');
```

```
for (int i=1; i+2 < s.length; i=i+4)
```

```
{
```

```
String t = s[i];
```

```
s[i] = s[i+2];
```

```
s[i+2] = t;
```

```
}
```

not off String op removed (read), off by one index is good

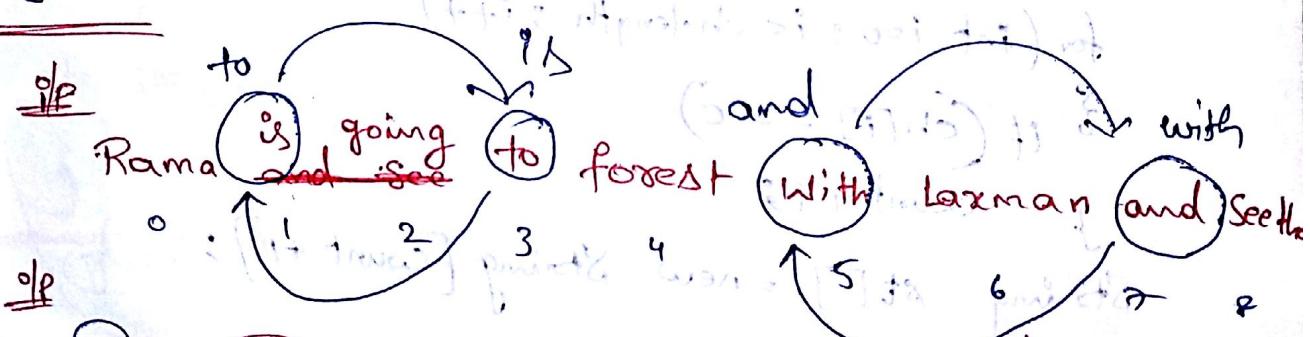
```
for (int i=0; i < s.length; i++)
```

```
{
```

```
res = res + s[i] + ":"; // add character to result string
```

```
SOP(res); // print result string
```

```
}
```



Rama to going is forest and laxman with Seetha

Q1) \rightarrow Rama is going to forest with Laxman and Seetha

Q2) Seetha is going to forest with Laxman and Rama

Q3) \rightarrow Rama is going to forest with Laxman and Seetha and

Q4) and is and to with forest Laxman going Seetha Ram

public class main

{ static String[] breakIt (String str, char c)

{ char ch[] = str.toCharArray();

int count = 0;

for (int i=0; i<ch.length; i++)

{ if (ch[i] == c)

count++; }

String st[] = new String [count+1];

int k=0;

st[k] = "";

for (int i=0; i<ch.length; i++)

{ if (ch[i] != c)

st[k] = st[k] + ch[i];

else

{ k++; st[k] = ""; }

return st;

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

```
{ Scanner sc1 = new Scanner (System.in);
```

```
SOP ("Enter your string");
```

```
String st = sc1.nextLine();
```

```
String s [] = breakIt (st, ' ');
```

If ($s.length \% 2 == 0$)

```
{ for (int i=0; i < s.length / 3 - 1; i += 2)
```

```
{
```

```
String t = s [i];
```

```
s [i] = s [s.length - i - 1];
```

```
s [s.length - i - 1] = t;
```

```
}
```

```
String res = "";
```

```
for (int i=0; i < s.length; i++)
```

```
{
```

```
res = res + s [i] + " ";
```

```
}
```

```
SOP (res);
```

```
}
```

~~for (int i=0; i < s.length / 3; i++)~~

~~else~~

```
{ for (int i=0; i < s.length / 4; i += 2)
```

```
{
```

```
String t = s [i];
```

```
s [i] = s [s.length - i - 1];
```

```
s [s.length - i - 1] = t;
```

```
}
```

* Find the middle character of middle word in a Sentence

→ Hai How are you
→ 0

public class Main;

{
 public (String[] args) {
 System.out.println("Hello");
 }

 Scanner scn1 = new Scanner(System.in);
 System.out.println("Enter string");

 String st = scn1.nextLine();

 String s[] = breakIt(st, " ");

 for (int i=0; i < s.length; i++)

 if (i == s.length/2)

 char ch[] = s[i].toCharArray();

 if (ch.length % 2 == 0)

 System.out.println(ch[ch.length/2] + ch[ch.length/2]);

 else

 System.out.println(ch[(ch.length/2)]);

 }

}

 if (i == s.length/2)

 if (i == s.length/2)

Variargs

(18) 903

(17) 903

* If Any method having arguments of same type differ by no. of parameters then we go for Variargs

* It is useful to decrease the length of the program

* method signature can have only one Variargs and it should be last argument.

int sum(double ... a); void m1(char ch, String str,
int ... a)
{
 int s=0;
 for(int i=0; i< args.length; i++)
 s=s+args[i];
 return s;
}

It represents Array
Array object
Array Name
int ... a

void m1(int ... a, char ch,
String str)

Package com.jspiders.cpl;

public class VarArgs

static int sum(int ... args)

static void method (String st, double b, char... ch)

{
 SOP(st);

 SOP(b);

 for (int i=0; i<ch.length(); i++)

 SOP(ch[i]);

PSUM (- - -)

{
 SOP("Hello World! and output of below is");

 SOP("*****");

 SOP("Sum of int Elements " + sum(10));

 SOP("Sum of int Elements " + sum(10, 20, 30, 40));

 SOP("method (" + "SITH", 97.50);

 SITH
 97.50

 method ("Praveen", 98.50, 76.50, 105.50);

Praveen
98.50
76.50
105.50

(*) ~~Read n number of strings~~ colours

and Display only unique colours

public class main

{

PSUM (- - - -)

Scanner scn1 = new Scanner(System.in);

String[] st =

SOP("Enter the num of ~~str~~ colours");

int n = scn1.nextInt();

String str[] = new String[n];

for (i=0; i<n; i++)

str[i] = scn.next();

int count;

for (i=0; i<n; i++)

{ if (count == 0)

{ for (j=i+1; j<n; j++)

{ if (str[i].equals(str[j]))

{ count++;

{ for (int k=j; k<n-1; k++)

{ str[k-1] = str[k];

}

n--;

j--;

}

{ if (count == 0) { sop(str[i]); }

for (int i=0; i<n; i++)

sop(str[i]);

}

= = = = =
 public static String[] colours (String[] str);
 { int n;
 int count;
 for (int i=0; i<n-1; i++)
 count=0;

for (j=i+1; j<n; j++)

{ if (str[i].equals(str[j]))

{ count++; } }

for (int k=j; k<n-1; k++)

{ str[k-1] = str[k];

if (count == 0)
sop(str[i]);