

3/7/19

1. Arithmetic operators: +, -, *, /, %
2. Relational operators: <, >, <=, >=, ==, !=
3. Logical operators: ||, &&, !

OR Operator

AND Operator

Exp1	Exp2	Result
T	X	T
T	T	T
F	T	F
F	F	F

Exp1	Exp2	Result
T	F	T
T	T	T
F	X	F
F	F	F

If first condition is true
then the result is true.

If first condition is false
then the result is false.

Eg 1. $4+3=7 \text{ } || \text{ } 4>5$

~~F || F~~

~~F and F~~

~~and and~~

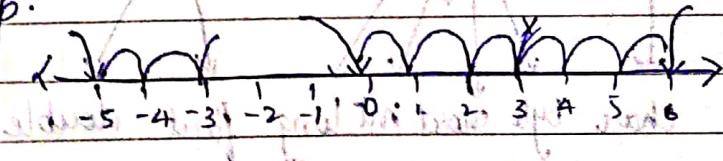
~~↓~~

2. $4!=4 \text{ } || \text{ } 5==5 \text{ } \&\& \text{ } 2*2=4 \Rightarrow \text{F} \text{ } || \text{ } \text{True} \Rightarrow \text{False} \text{ } \&\& \text{True} = \text{True}$

~~2*2=4~~ $\&\& \text{ } 3*2=6 \text{ } \&\& \text{ } 4*1>=4 \Rightarrow \text{True} \&\& \text{True} = \text{True}$

~~3*2=6~~ $\&\& \text{ } 4*1>=4 \Rightarrow \text{True} \&\& \text{True} = \text{True}$

* For loop.



for (int i = -5; i <= -3; i++) for (int i = 0; i <= 3; i++) for (int i = 6; i >= 3; i--)

~~{~~ S.O.P('*'); }

S.O.P('*'); }

S.O.P('*'); }

* Printing a ^a_b_c d

Sol: class Demo

{

public static void main (String args[])

{

System.out.print ("a");

System.out.print ("d");

System.out.print ("b");

System.out.print ("c");

}

}

Data Types

Primitive

non-Primitive

- Class

- Array

- Interface

- String

Boolean & Character

Numeric

boolean char byte short int long float double

(T/F)

It also occupies some memory but the

memory allocated depends on OS that is being used.

21/2/2019

* Precedence chart:

1. Arithmetic operations

* , / highest precedence, left to right associative

+, - low precedence, left to right associative

2. Logical operations

&& highest precedence

|| lower precedence

3. () - Parenthesis have the highest precedence among all the operations

* Pattern Programming

O/P: * * * * * * * * * *
 rows (i)
 columns (j)

i	j				
0	0	1	2	3	4
1	0	1	2	3	4
2	0	1	2	3	4
3	0	1	2	3	4
4	0	1	2	3	4

```

class Pat1 {
    public static void main (String args[]) {
        for (int i = 0; i <= 4; i++) {
            for (int j = 0; j <= 4; j++) {
                System.out.print ("*");
            }
            System.out.println ();
        }
    }
}

```

Output: *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

* * * *

	0	1	2	3	4
2	*	*	*	*	*
1	*			*	
0	*		*		
4	*	*	*	*	*

Ques 2 class Pat 2.

{

public static void main (String args[])

{

for (int i=0; i<=4; i++)

{

 for (int j=0; j<=4; j++)

{

 if ((i==0 || i==4 || j==0 || j==4))

{

 System.out.print ("*");

{

else

 System.out.print (" ");

{

 System.out.println ();

{

 if (i!=0 & i!=4 & j!=0 & j!=4)

{

 System.out.print (" ");

{

 System.out.println ();

{

 System.out.print (" ");

{

 System.out.println ();

{

 System.out.print (" ");

{

 System.out.println ();

{

3... User defined input for previous program.

```

import java.util.Scanner;
class Pat3
{
    public static void main (String args[])
    {
        Scanner scnr = new Scanner (System.in);
        System.out.print ("Enter the limit : ");
        int n = scnr.nextInt();
        for (int i=0; i<n; i++)
        {
            for (int j=0; j<n; j++)
            {
                if ((i==0) || (i==n-1) || (j==0) || (j==n-1))
                    System.out.print ("*");
                else
                    System.out.print (" ");
            }
            System.out.println ();
        }
    }
}
  
```

O/P: Enter the limit : 3

```

* * *
*   *
* * *
  
```

H. ~~$n/4 \rightarrow n/2 \rightarrow 3n/4$~~

$n/4$	$n/2$	$3n/4$	O/P: * * * * * * * *
*	*	*	* * * * *
*	*	*	* * * * *
*	*	*	* * * *
*	*	*	* * *
*	*	*	* * *
*	*	*	* * *
*	*	*	* * *

Logic: if ($i == 0 || i == n-1 || i == n/2 || i == n/4 || i == 3n/4 ||$
 $j == 0 || j == n-1 || j == n/2 || j == n/4 || j == 3n/4;$)

Sol- import java.util.Scanner;
 class Pat4

{
 public static void main (String args [])

Scanner scan = new Scanner (System.in);
 System.out.print ("Enter the limit:");
 int n = scan.nextInt();

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

{
 for (j = 0; j < n; j++)

if ($i == 0 || i == n-1 || i == n/4 || i == n/2 || i == 3n/4 ||$
 $j == 0 || j == n-1 || j == n/4 || j == n/2 || j == 3n/4)$

{
 System.out.print ("*");

else

{
 System.out.print (" ");

{
 System.out.println ();

5. Pattern A

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

```
class Pat A
```

```
{
```

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

```
{
```

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

```
    System.out.print ("Enter the limit:");
```

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

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

```
{
```

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

```
{
```

```
            if (i==0 || i==n/2 || j==0 || j==n-1)
```

```
{
```

```
                System.out.print ("*");
```

```
}
```

```
            else
```

```
{
```

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

```
}
```

```
{
```

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

```
}
```

```
{
```

O/p: Enter the limit : 5

```
* * * * *
```

```
* * * * *
```

```
* * * * *
```

```
*
```

```
* *
```

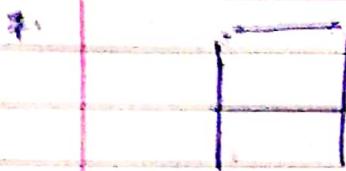
6. O/P:



```

import java.util.Scanner;
class PatA
{
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit:");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((i == 0 && j < n/2) || (i == n/2 && j < n/2) || (j == 0) || (j == n/2)
                    || (i + j == n/2) || (i + j == n - 1) || (i == j))
                    System.out.print ("*");
                else
                    System.out.print (" ");
            }
            System.out.println ();
        }
    }
}
  
```

O/P: * * *
* * *
* * *
* * *
* * *



Ex. import java.util.Scanner; *(for Java 8 and later)*

class BoxA2

{ *(for public class)* main method added

public static void main (String args[])

{ *(for main method)* scanner object created

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{ *(for printing pattern)*

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

if ((j == 0 && i == 0) || (i == 0 && j == 0) || (j < n/2) &&

(i == n/2 && j < n/2) || (j == n/2 && i == 0))

{ *(for printing pattern)*

System.out.print ("*");

else { *(for printing blank space)*

{ *(for printing blank space)*

System.out.print (" "));

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

* * * * * *(for printing blank space)*

O/P: Enter the limit : 5

3 * * * *

3 * * * *

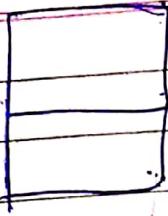
3 * * * *

3 * * * *

3 * * * *

3 * * * *

8.



```
8. import java.util.Scanner;
class PatB
```

{

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

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

```
System . out . print ( " Enter the limit " ) ;
```

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

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

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

```
if ( i == 0 && j < n / 2 - 1 ) {
```

```
j = n / 2 && j >= n / 2 || j == 0 ||
```

```
j == n / 2 || i == n - 1 && j < n / 2 ) {
```

```
(* *) true; two - matrix
```

```
System . out . print ( "*" ) ;
```

```
} else {
```

```
(* *) true; two - matrix
```

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

```
} }
```

O/P: Enter the limit : 5

```
* * (*) true; two - matrix * * *
```

```
* system . out . println ( ) ; *
```

```
* * *
```

```
* * *
```

```
* *
```

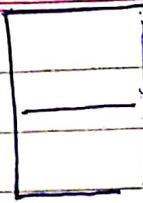
```
* * *
```

```
* *
```

```
* *
```

Scanned with CamScanner

8.



Sol -> import java.util.Scanner;
class PatB1

{

public static void main (String args [])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt ();

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

{

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

{

if ((j == 0 || j == n / 2) && (i == 0 || i == n / 2) || (i == n - 1))

else if ((i == 0 || i == n / 2) || (j == n - 1) && (j < n / 2))

System.out.print ("*");

} else

{

System.out.print (" "));

} }

Q: What will print?

O/P: Enter the limit 7

* * * * * * * * *

{ * * * * *

* * * * *

* * * * *

* * * * *

* * * * *

* * * * *

9

```
import java.util.Scanner;
class Patc
```

{

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

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

```
System.out.print ("Enter the limit");
```

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

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

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

```
if ((i * j) % 10 == 0 || (i * j) == (n - 1) / 2 || j == 0)
```

```
ans++; if (ans > 1) System.out.print ("*"); else
```

```
System.out.print ("*"); } System.out.println (); }
```

```
else
```

```
{ int ans = 0; if (i * j == 0 || (i * j) % 10 == 0)
```

```
System.out.print ("*"); } System.out.println (); }
```

O/p: Enter the limit 7

0 1 2 3 4 5 6 7

System.out.println ();

0 1 2 3 4 5 6 7

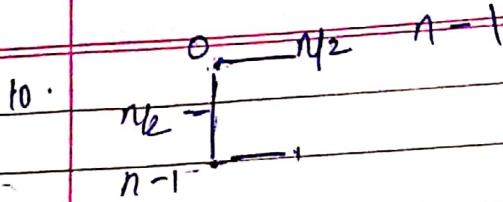
2: 5 6 7

3: * * * 0 1 2 3 4 5 6 7

* 0 1 2 3 4 5 6 7

* 0 1 2 3 4 5 6 7

* * * * 0 1 2 3 4 5 6 7



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

$i = 0 \text{ & } j = 0 \text{ & } j! = 0$

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

import java.util.Scanner;

class Pat1

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if ((j==0 && i==0) && i!=n-1 || i==0 && j==0
|| i==n-1 && j==0)

{

System.out.print ("*");

O/P: Enter the

limit : 5

else

System.out.print (" "));

}

System.out.println();

*

*

11.



```

import java.util.Scanner;
class Pat D
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the limit:");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((j == 0 || j == n / 2) || (i == 0 || i == n - 1) || (i == 0 || i == n - 1) && j < n / 2)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}

```

11.

E

Ques → import java.util.Scanner;
class Pat

{

public static void main(String args[])

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if (j == 0 || (i == 0 || i == n / 2 || i == n - 1) && j < n / 2)

{

System.out.print("*");

}

else

System.out.print(" ");

}

System.out.println();

}

}

O/P: Enter the limit: 7

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

13

```

So -> import java.util.Scanner;
class PatF
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the limit");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((j == 0) || (i == 0) || (i == n / 2) && j < n / 2)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}

```

O/P: Enter the limit : 7

```

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

```

14.

```
Set-> import java.util.Scanner;  
class Path {
```

{

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

{

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

```
System.out.print ("Enter the limit");
```

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

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

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

```
        if ((j == 0 || i == n / 2 && j < n / 2) || j == n / 2)
```

```
            System.out.print ("*");
```

```
        else
```

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

```
    }
```

```
    System.out.println ();
```

```
}
```

```
}
```

```
    System.out.println ();
```

```
}
```

O/P: Enter the limit : 7

* * *

* * * *

* * *

* * *

* *

*

*

15.

```
Sol-> import java.util.Scanner;
class Pat I
```

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

```
Scanner scan = new Scanner (System.in);
System.out.print ("Enter the limit");
int n=scan.nextInt();
```

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

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

```
(if (i==0 || i==n) && j<n/2) || j==n/4)
{
```

```
System.out.print ("*");
}
```

```
else
```

```
print System.out.print (" ");
}
```

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

O/P: Enter the limit: 4

* * * *

*

*

*

*

* * *

16.

```
Sol) import java.util.Scanner;  
class Pat {  
    public static void main (String args[]) {  
        Scanner scan = new Scanner (System.in);  
        System.out.print ("Enter the limit");  
        int n = scan.nextInt();  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n; j++) {  
                if ((i == 0 && (j < n/2)) || (j == n/4) || (i == n-1 && j < n/4))  
                    System.out.print ("*");  
                else  
                    System.out.print (" ");  
            }  
            System.out.println ();  
        }  
    }  
}  
O/P: Enter the limit: 7  
* * * * * * *  
* * * * * * *  
* * * * * * *  
* * * * * * *  
* * * * * * *  
* * * * * * *
```

17

```
sol-> import java.util.Scanner;
```

```
class Part 1
```

{

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

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

```
System.out.print("Enter the limit");
```

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

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

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

```
{ if (j == 0 || i == n - 1 || j < n / 2)
```

```
System.out.print("*");
```

}

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

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

}

O/P: Enter the limit : 7

*

*

*

*

*

18.

1

Q1. -> import java.util.Scanner;
class Pat0

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if ((j == 0 && i == 0) || i == n - 1 ||

j == n / 2 || i == n - 1 && j == 0 ||

(i == 0 || i == n - 1) && j < n / 2 && j != 0)

{

System.out.print ('*');

}

else

{

System.out.print (' ');

}

System.out.println ();

}

}

O/P: Enter the limit : 5

*

* *

* *

* *

*

19.

F

Sol → import java.util.Scanner;
class PartP;

{

public static void main (String args[])

{

Scanner scbm = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scbm.nextInt();

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

{

for (int j=0; j<5; j++) {

{

If (j==0 || j==n/2 || i==0 || i==n/2 ||

(i<n/2 & j==0) || (i==n/2 & j<n/2) || j==n/2)

{

System.out.print ("*");

}

else

System.out.print (" ");

}

System.out.println ();

}

}

O/P: Enter the limit: 5

* * * * *

* * * *

* * * *

* * * *

* * * *

20.

```
Q1 => import java.util.Scanner;
class Pats
```

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

```
Scanner scan = new Scanner (System.in);
System.out.print ("Enter the limit");
int n = scan.nextInt();
for (int i = 0; i < n; i++)
{
```

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

```
if ((i == 0 || i == n/2 || i == n-1) && j <= n/2)
    if (j == 0 && i < n/2 || j == n/2 && i > n/2)
{
```

```
    System.out.print ("*");
}
else
    System.out.print (" ");
}
```

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

{

B

O/p: Enter the limit : 5

*

*

Q1.

Q1) `import java.util.Scanner;`
 class PatT

{

public static void main(String args[])

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if (i==0 || j<n/2 || j==n/4)

{

System.out.print("*");

{

else

{

System.out.print(" "));

{

System.out.println();

{

{

{

O/P: Enter the limit: 7

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

22.



Sol -> import java.util.Scanner;

class PatV.

{

public static void main(String args[])

{

Scanner scan = new Scanner(System.in);

System.out.print("Enter the limit");

int n = scan.nextInt();

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

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

if (j == 0 && i == n - 1 || j == n / 2 && i == n - 1 ||

i == n - 1 && j == 0 && i == n - 1 && j == n / 2)

{

System.out.print("*");

{

else

{

}

}

}

}

}

}

}

}

}

}

O/P: Enter the limit: 7

* *

* *

* *

* *

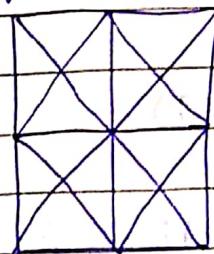
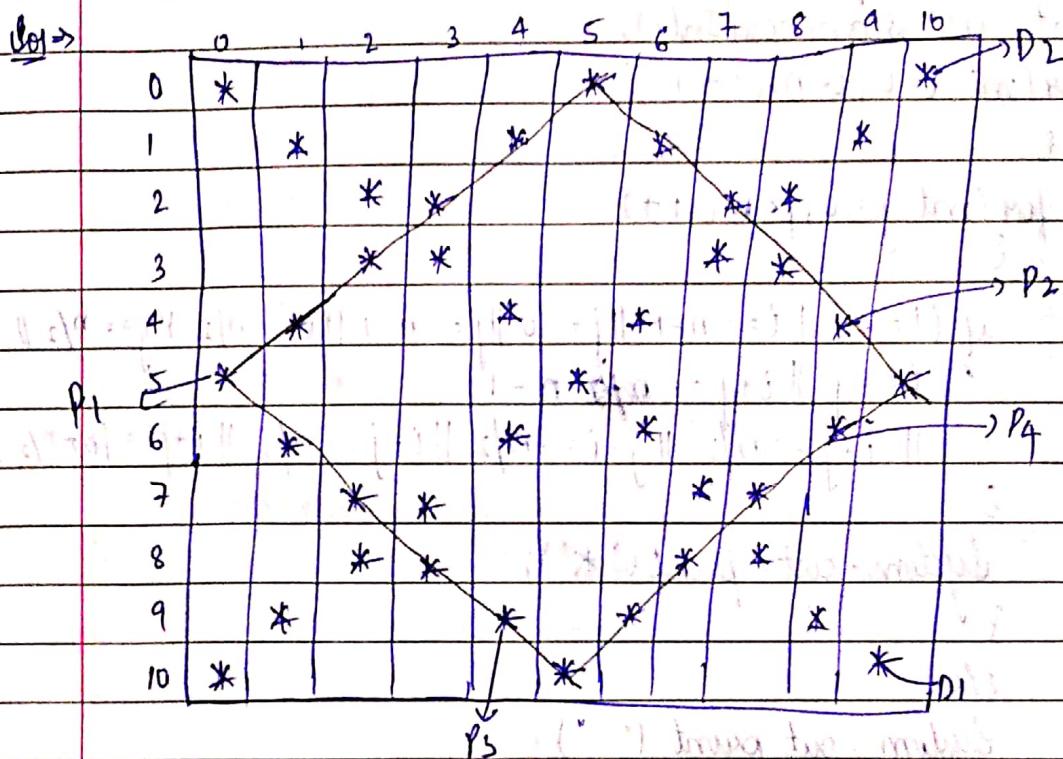
* *

* *

* *

* Rangoli Patterns.

Q3

 \Rightarrow 

D_1	D_2	P_1	P_2	P_3	P_4
(0, 0)	(10, 10)	(0, 5)	(0, 5)	(5, 0)	(5, 10)
(1, 1)	(1, 9)	(1, 4)	(1, 6)	(6, 1)	(6, 9)
(2, 2)	(2, 8)	(2, 3)	(2, 7)	(7, 2)	(7, 8)
(3, 3)	(3, 7)	(3, 2)	(3, 8)	(8, 3)	(8, 7)
:	:	(4, 1)	(4, 9)	(9, 4)	(9, 6)
(10, 10)	(10, 0)	(5, 0)	(5, 10)	(10, 5)	(5, 5)

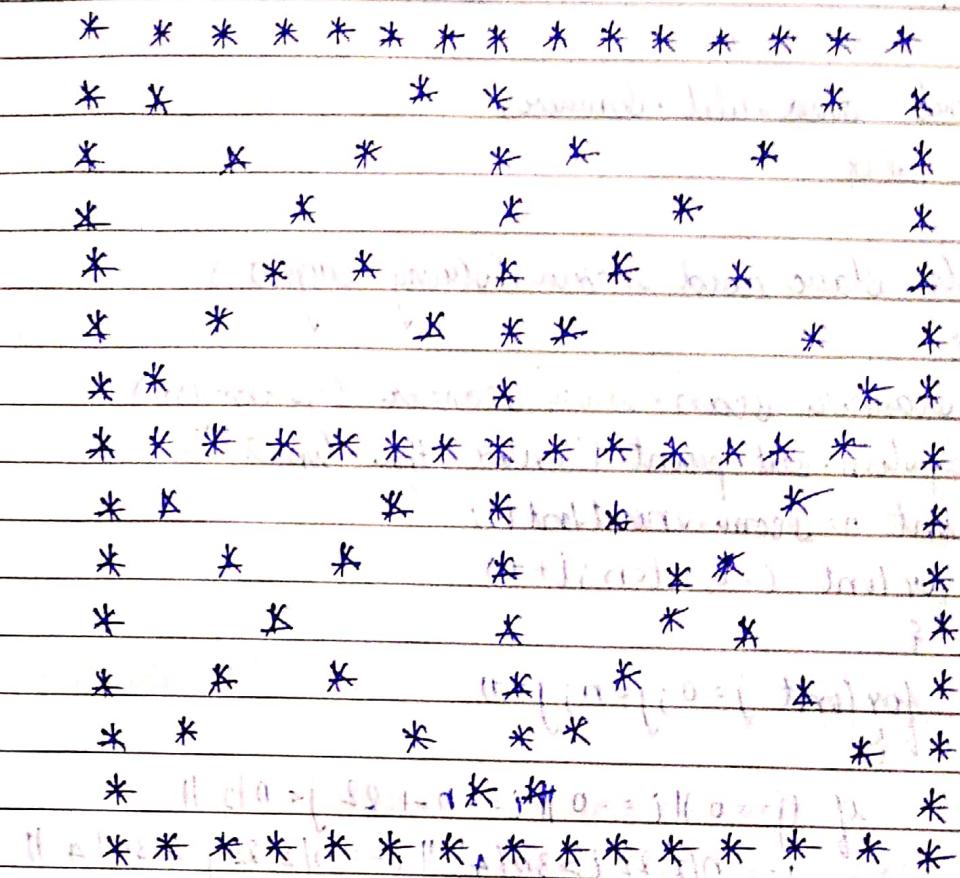
$$i=j \quad i+j=n-1 \quad i+j=\frac{n}{2} \quad j-i=\frac{n}{2} \quad i-j=\frac{n}{2} \quad i+j = \frac{(n+n)}{2}-1$$

Big diagonals

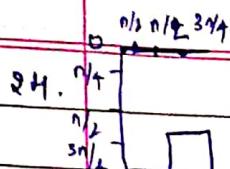
Square sides.

```
import java.util.Scanner;  
class PatLangoli  
{  
    public static void main(String args[])  
    {  
        Scanner scan = new Scanner(System.in);  
        System.out.print("Enter the limit:");  
        int n = scan.nextInt();  
        for (int i = 0; i < n; i++)  
        {  
            for (int j = 0; j < n; j++)  
            {  
                if (i == 0 || i == n - 1 || j == 0 || j == n - 1 || i == n / 2 || j == n / 2 ||  
                    i == j || i + j == n - 1 || i + j == n / 2 || j - i == n / 2 || i - j == n / 2 || i + j == (n + n / 2) - 1)  
                {  
                    System.out.print("*");  
                }  
                else  
                {  
                    System.out.print(" ");  
                }  
            }  
            System.out.println();  
        }  
    }  
}
```

O/P: Enter the limit: 15



(Ans is 15)



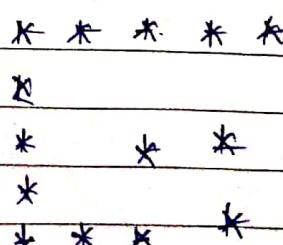
```

24. n
    0 1/2 n/2 3n/4

import java.util.Scanner;
class PalG
{
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((j == 0) || (i == 0) || (i == n - 1) || (j < n / 2) ||
                    (j == n / 2) || (j > 3n / 4) || (i == n / 2) || (j > 3n / 4) ||
                    (j == n - 1) || (i > 3n / 4))
                {
                    System.out.print ("*");
                }
                else
                {
                    System.out.print (" ");
                }
            }
            System.out.println ();
        }
    }
}

```

O/P: Enter the limit : 5



Q5.



```
import java.util.Scanner;
class PatG1
```

{

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

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

```
System.out.print("Enter the limit");
```

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

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

```
{
```

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

```
{
```

```
if (j == 0 && i == 0 && i == n - 1 || i == 0 && j == 0 ||
```

```
i == n - 1 && j < n / 2 && j == 0 || j == n / 2 && i > 3 * n / 4 ||
```

```
i == 3 * n / 4 && j > n / 2 || j == n - 1 && i > 3 * n / 4 )
```

```
{
```

```
System.out.print("*");
```

```
}
```

```
else
```

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

```
}
```

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

```
}
```

```
}
```

```
O/P:
```

Enter the limit : 7

Scanned with CamScanner

26

```
(let) import java.util.Scanner;
class Palindrome {
    }
```

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

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

```
System.out.print ("Enter the limit");
```

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

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

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

```
if (j == 0 && i < 3 * n / 4 || i == 0 && j < n / 2 + 1) {
    }
```

```
if (j < n / 2 && i == 3 * n / 4 || j >= n / 2 && i < 3 * n / 4) {
    }
```

```
(if (i < j && i != j) && j > n / 2) {
    }
```

```
    }
```

```
    System.out.print ("*");
}
```

```
else
    System.out.print (" ");
}
```

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

```
3 * * * *
```

```
3 * * * *
```

```
* * * * *
```

```
* * * *
```

```
* * * *
```

```
* * * *
```

```
* * * *
```

27.

17

Sol) import java.util.Scanner;
class Pat {

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if (j == 0 && i < 3 * n / 4 && i != 0 - 2 * j) {

if (i == 0 && j == 3 * n / 4 && j != 0 && j < 3 * n / 4)

if (i == 3 * n / 4 && j < 3 * n / 4 && j != 0)

j == 3 * n / 4 && i < 3 * n / 4 && i != 0)

i == j && j > n / 2)

{

System.out.print ("*");

{

System.out.print (" ");

{

System.out.println ();

{

O/P: Enter the limit.

{

Q8.

K

```

import java.util.Scanner;
class Park
```

```

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

```

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

```

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

```

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

```

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

```

{
```

```

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

```

{
```

```

if (j == 0 || i + j == n / 2 || i - j == n / 2)
```

```

{
```

```

System.out.print ("*");
```

```

{
```

```

else
```

```

{
```

```

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

```

}
```

```

System.out.println ();
```

```

}
```

O/P: Enter the limit: 7

```

*****
 * *
 * *
 *
 * *
 * *
 * *
```

29.

```
↳ import java.util.Scanner;
```

```
class PartM
```

```
{
```

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

```
{
```

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

```
System.out.print ("Enter the limit");
```

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

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

```
{
```

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

```
{
```

```
if ((j == 0 || j == n - 1) || (i == j || i + j == n - 1) || (i == n / 2))
```

```
System.out.print ("*");
```

```
}
```

```
else
```

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

```
}
```

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

```
}
```

```
}
```

O/P: Enter the limit : 7

```
* * * * * * *
```

```
* * * * * * *
```

```
* * * * * * *
```

```
* * * * * * *
```

```
* * * * * * *
```

```
* * * * * * *
```

30.

N

```

sol-> import java.util.Scanner;
class PatN {
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit:");
        int n = scan.nextInt();
        for (int i=0; i<n; i++)
        {
            for (int j=0; j<n; j++)
            {
                if ((j==0 || j==n-1 || i==j) ||
                    (i>0 & i<n-1 & j>i+1) || (i>n-1 & j!=0))
                    System.out.print ("*");
                else
                    System.out.print (" ");
            }
            System.out.println ();
        }
    }
}

```

O/P: Enter the limit: 7

```

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

```

3.

Sol: `import java.util.Scanner;`

along Part R

{

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

{

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

`System.out.print ("Enter the limit");`

`int n = scan.nextInt();`

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

{

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

`if (j == 0 || i == 0 && j < n/2 || j == n/2 && i < n/2 ||`

`i == n/2 && j <= n/2 || i == j && i > n/2)`

{

`System.out.print ("*");`

{

`else`

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

{

`System.out.println ();`

{

{

O/P: Enter the limit : 5

* * *

* * *

* * *

* * *

* * *

32.

```
import java.util.Scanner;  
class PatV  
{  
    public static void main(String args[])  
    {  
        Scanner scan = new Scanner(System.in);  
        System.out.print("Enter the limit: ");  
        int n = scan.nextInt();  
        for (int i = 0; i < n; i++)  
        {  
            for (int j = 0; j < n; j++)  
            {  
                if ((i == j) || (j == n - 1) || (i == n / 2))  
                {  
                    System.out.print('*');  
                }  
                else  
                {  
                    System.out.print(' ');  
                }  
            }  
            System.out.println();  
        }  
    }  
}
```

O/p: Enter the limit: 9

```
* * * * * * * * *  
* * * * * * * * *  
* * * * * * * * *  
* * * * * * * * *  
* * * * * * * * *
```


34.

X

Sol:

```
import java.util.Scanner;  
class PartV  
{  
    public static void main(String args[]){  
        Scanner Scam = new Scanner(System.in);  
        System.out.print("Enter the limit");  
        int n = Scam.nextInt();  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n; j++) {  
                if ((i == j) || (i + j == n - 1))  
                    System.out.print("*");  
                else  
                    System.out.print(" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

O/P: Enter the limit: 5

* * * * *

* * * * *

* * * * *

* * * * *

35.

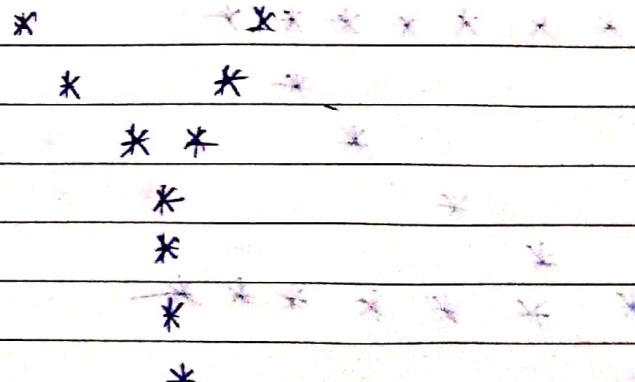
Y

```

So, > import java.util.Scanner;
class Party
{
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((i == j) || (i + j == n - 1) && i <= n / 2) || (j == n / 2) || (i > n / 2))
                    System.out.print ("*");
                else
                    System.out.print (" ");
            }
            System.out.println ();
        }
    }
}

```

O/P: Enter the limit : 7



36.

```
↳ import java.util.Scanner;  
class PatZ
```

{

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

{

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

```
System.out.print ("Enter the limit");
```

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

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

{

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

{

```
if ((i == 0 || i == n - 1) || (i + j == n - 1))
```

{

```
System.out.print ("*");
```

{

```
else
```

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

{

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

{

{

O/P: Enter the limit : 7

```
* * * * *
```

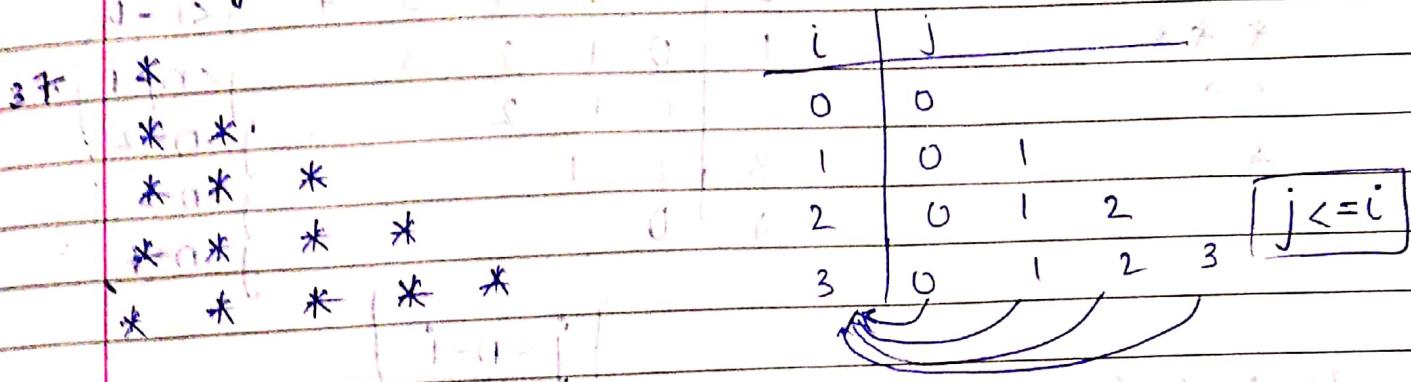
```
* * *
```

```
*
```

```
*
```

```
* * * * *
```

Triangle Pattern



```
So, import java.util.Scanner;
class Triangle {
    ?
```

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

```
Scanner scan = new Scanner(System.in);
System.out.print("Enter the limit");
int n = scan.nextInt();
for (int i = 0; i < n; i++) {
    ?
}
```

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

```
System.out.print("*");
}
```

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

O/P: Enter the limit: 3

```
* * * *
* * *
* *
```

38.

	i	j					
*	0	0 1 2 3 4					$j < n - i$
**	1	0 1 2 3					$j < n - 1$
***	2	0 1 2					$j < n - 2$
****	3	0 1					$j < n - 3$
*****	4	0					$j < n - 4$

$j < n - i$

Sol: import java.util.Scanner;
 class Triangle2
{

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

```
Scanner scan = new Scanner (System.in);
System.out.print ("Enter the limit");
int n = scan.nextInt();
for (int i = 0; i < n; i++)

```

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

```

```
System.out.print ("*");
}
```

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

```
}
```

O/P: Enter the limit: 4

```
*****
****
 ***
 **
 *
```

39. Trick 1: Printing patterns horizontally in one line.

We should have one i loop and inside i we should have multiple j loops.

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

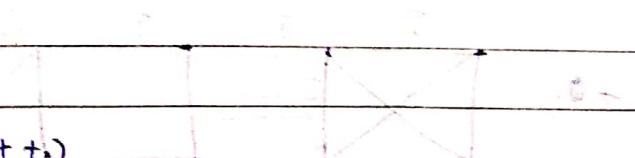
`for (int j=0; j<n; j++)`

`{`

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

`{`

`}`



④

```
* * * $ $ $  
* * * $ $ $  
* * * $ $ $
```

Output:
`import java.util.Scanner;`
`class Pattern3`
`{`

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

`{`

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

`System.out.print ("Enter the limit");`

`int n = scan.nextInt();`

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

`{`

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

`{`

`System.out.print ("*");`

`}`

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

first row in flattened position. I start

System.out.println("\$");

Internally it is known that you have stored character.

System.out.println();

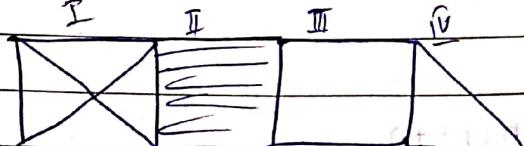
(\leftrightarrow char is printed as it is)

}

2.

\leftrightarrow char is printed as it is

40.



\leftrightarrow char is printed as it is

{ }

{ }

if else if
if else if
if else if

conversion - string and
datatype

String format - how adds nothing

(Conversion happens automatically)

(String with printf) format - two methods

(1) to print max - n long

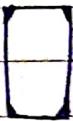
(2) to print float - printf - float

format - \leftrightarrow (printf - float)

format - \leftrightarrow (% float - float)

Numbers

44



Q1 -> import java.util.Scanner
class Num0

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{

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

{

if (j == 0 && i == 0 && i == n - 1 || j == n / 2 && i == n - 1 && i == 0
|| (i == 0 || i == n - 1) && j < n / 2 && j != 0)

{

System.out.print ("*");

}

else

System.out.print (" ");

}

System.out.println ();

}

}

O/p: Enter the limit: 5

*
* *
* *
* ***

4.5. Num1

```
import java.util.Scanner;
class Num1
```

{

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

{

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

```
System.out.print ("Enter the limit : ");
```

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

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

{

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

{

```
If (j == 0) scan.nextInt();
```

{

```
System.out.print ('*');
```

{

else

```
System.out.print (' ');
```

{

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

{

O/P: Enter the limit : 5

```
*****
* * * * *
*       *
*   *   *
*     *
*   *   *
```

45.

```

↳ import java.util.Scanner;
class Num2
{
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit");
        int n = scan.nextInt();
        for (int i=0; i<n; i++)
        {
            for (int j=0; j<n; j++)
                if ((i==0 || j==n-1) || i<n/2 || i==n/2 || i==n-1
                   || j==0 || i>n/2)
                    System.out.print ("*");
                else
                    System.out.print (" ");
            System.out.println ();
        }
    }
}

```

O/P: Enter the limit: 5

```

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

```

44

```
Ques import java.util.Scanner;  
class Num3  
{  
    public static void main(String args[])  
    {  
        Scanner scan = new Scanner(System.in);  
        System.out.print("Enter the limit");  
        int n = scan.nextInt();  
        for (int i = 0; i < n; i++)  
        {  
            for (int j = 0; j < n; j++)  
            {  
                if ((i == 0 || j == n - 1) || i == n / 2 || j < n - 1  
                    || j == 0 || i > n - 1 || i == n - 1)  
                {  
                    System.out.print("*");  
                }  
                else  
                {  
                    System.out.print(" ");  
                }  
            }  
            System.out.println();  
        }  
    }  
}
```

O/p: Enter the limit: 5

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

MS.

4

Ques.

import java.util.Scanner;

class Num4

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

{

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

{ if ((j == 0 || i < n/2 || i == n/2 || j == n-1) ||

System.out.print ("*");

{

else

{

System.out.print (" ");

System.out.println ();

}

{

}

O/P: Enter the limit: 5

* * * * * : formed all return

* * * * * *

* * * * * *

* * * * * *

* * * * * *

49.

```

import java.util.Scanner;
class Num5
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the limit:");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((i == 0 || i == n / 2 || i == n - 1) && j < n / 2 ||
                    j == 0 && (j < n / 2 || j == n / 2) || j >= n / 2)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}

```

O/p: Enter the limit: 5

```

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

```

52.



```

So1> import java.util.Scanner;
class Num8
{
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter the limit:");
        int n = scan.nextInt();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if ((i == 0 || j == 0 || i == n/2 || i == n-1 || j == n-1)
                    || (i + j == n - 1))
                    System.out.print ("*");
                else
                    System.out.print (" ");
            }
            System.out.println ();
        }
    }
}

```

O/P: Enter the limit: 5

```

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

```

53



Ques import java.util.Scanner;
class Num9

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the limit");

int n = scan.nextInt();

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

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

if (j == 0 && i < n/2 || i == 0 && j < n/2 || j == n/2
|| i == n/2 && j <= n/2)

System.out.print ("*");

else

System.out.print ("_");

System.out.println ();

O/P : Enter the limit : 5

* * *

* *

* * *

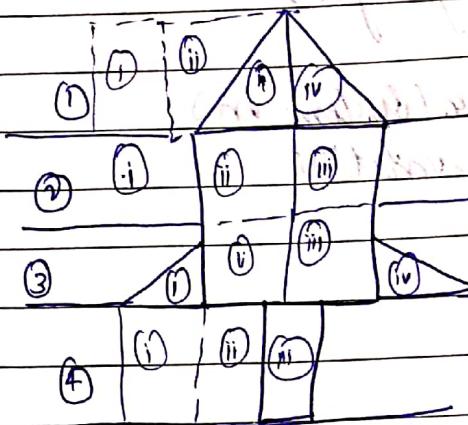
*

*

54. Truck 2: Painting vertically



55.



Painting vertical rows from right to left

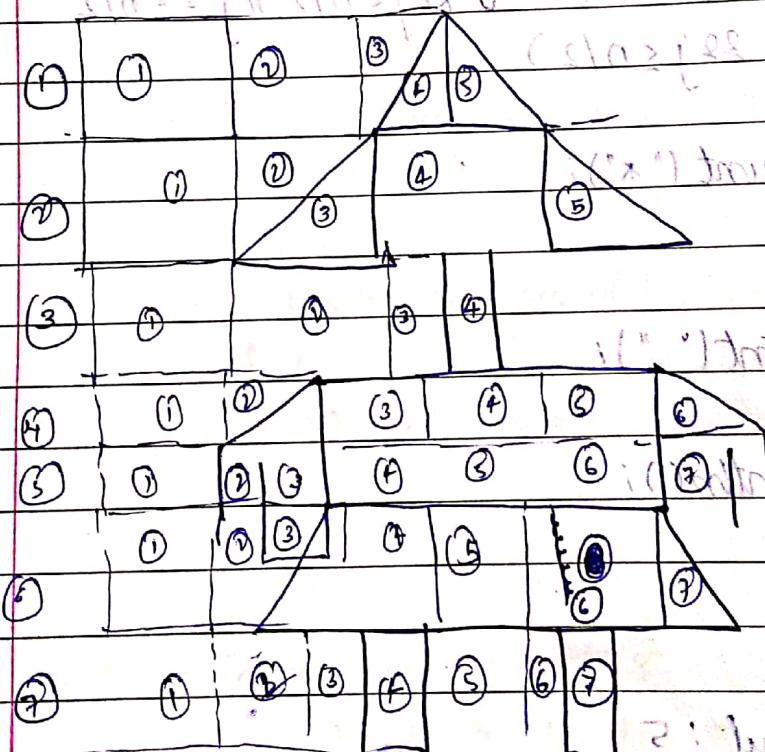
(e.g., paint from bottom up)

(1) Paint from right to left
(2) Paint from right to left

(3) Paint from right to left

(4) Paint from right to left

56.



$\{1 \leq a \leq 11 \wedge 1 \leq b \leq 11\} \rightarrow 11 \cdot 11 = 121$

$\{1 \leq a \leq 12 \wedge 1 \leq b \leq 12\} \rightarrow 12 \cdot 12 = 144$

$\{1 \leq a \leq 13 \wedge 1 \leq b \leq 13\} \rightarrow 13 \cdot 13 = 169$

$\{1 \leq a \leq 14 \wedge 1 \leq b \leq 14\} \rightarrow 14 \cdot 14 = 196$

$\{1 \leq a \leq 15 \wedge 1 \leq b \leq 15\} \rightarrow 15 \cdot 15 = 225$

$\{1 \leq a \leq 16 \wedge 1 \leq b \leq 16\} \rightarrow 16 \cdot 16 = 256$

$\{1 \leq a \leq 17 \wedge 1 \leq b \leq 17\} \rightarrow 17 \cdot 17 = 289$

$\{1 \leq a \leq 18 \wedge 1 \leq b \leq 18\} \rightarrow 18 \cdot 18 = 324$

$\{1 \leq a \leq 19 \wedge 1 \leq b \leq 19\} \rightarrow 19 \cdot 19 = 361$

$\{1 \leq a \leq 20 \wedge 1 \leq b \leq 20\} \rightarrow 20 \cdot 20 = 400$

$\{1 \leq a \leq 21 \wedge 1 \leq b \leq 21\} \rightarrow 21 \cdot 21 = 441$

$\{1 \leq a \leq 22 \wedge 1 \leq b \leq 22\} \rightarrow 22 \cdot 22 = 484$

$\{1 \leq a \leq 23 \wedge 1 \leq b \leq 23\} \rightarrow 23 \cdot 23 = 529$

$\{1 \leq a \leq 24 \wedge 1 \leq b \leq 24\} \rightarrow 24 \cdot 24 = 576$

$\{1 \leq a \leq 25 \wedge 1 \leq b \leq 25\} \rightarrow 25 \cdot 25 = 625$

$\{1 \leq a \leq 26 \wedge 1 \leq b \leq 26\} \rightarrow 26 \cdot 26 = 676$

$\{1 \leq a \leq 27 \wedge 1 \leq b \leq 27\} \rightarrow 27 \cdot 27 = 729$

$\{1 \leq a \leq 28 \wedge 1 \leq b \leq 28\} \rightarrow 28 \cdot 28 = 784$

$\{1 \leq a \leq 29 \wedge 1 \leq b \leq 29\} \rightarrow 29 \cdot 29 = 841$

$\{1 \leq a \leq 30 \wedge 1 \leq b \leq 30\} \rightarrow 30 \cdot 30 = 900$

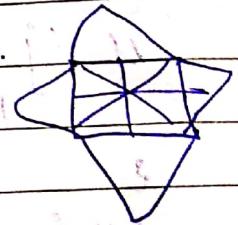
$\{1 \leq a \leq 31 \wedge 1 \leq b \leq 31\} \rightarrow 31 \cdot 31 = 961$

$\{1 \leq a \leq 32 \wedge 1 \leq b \leq 32\} \rightarrow 32 \cdot 32 = 1024$

$\{1 \leq a \leq 33 \wedge 1 \leq b \leq 33\} \rightarrow 33 \cdot 33 = 1089$

$\{1 \leq a \leq 34 \wedge 1 \leq b \leq 34\} \rightarrow 34 \cdot 34 = 1156$

$\{1 \leq a \leq 35 \wedge 1 \leq b \leq 35\} \rightarrow 35 \cdot 35 = 1225$



57.



58.



Number Pattern

	1	2	3	4
59.1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4

Sch

~~10000-3000~~ (Total net)

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

三

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

5

Training: $\frac{1}{2} \times \text{Actual}$

? System.out.println(i);

3

Comparing two methods

System.out.println();

3

卷之三

卷之三

卷之三

1975
June

	1	2	3	4
60.1	1	2	3	4
2	1	2	3	4
3	1	2	3	4
4	1	2	3	4

Soln

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

```
{
```

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

```
{
```

```
        System.out.print(j);
```

```
}
```

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

```
}
```

```
}
```

O/P: 12345

12345

12345

12345

12345

61. 60 program with spaces.

Ways of writing 3
 1. white space 2. white space 3.
 1. white space 2. white space 3.
 1. white space 2. white space 3.

Ans

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

{

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

{

 System.out.print(j);

 System.out.print(" ");

}

 System.out.println();

}

{

 ((i+1) * (n - i)) * ((n - i) + 1)

 ((i+1) * (n - i)) * ((n - i) + 1)

 ((i+1) * (n - i)) * ((n - i) + 1)

62.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16.

here i and j values are not
related so we need
to take an extra variable

Sol.) ~~for~~ int c = 1;

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

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

 System.out.print(c + " ");

used to increment the value
after printing first.

System.out.println();

63. 1 * 2 * 3 * 4

5 * 6 * 7 * 8

9 * 10 * 11 * 12

13 * 14 * 15 * 16.

int c = 1;

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

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

 if (c > 0) System.out.print(c + " ");

 if (j < n)

 System.out.print("*");

}

 System.out.println();

y

J

64. 16 15 14 13 9 8 7 6 4 3
 12 11 10 9 or 6 5 4 or 2 1
 8 7 6 5 11 3 2 1 10 9 8 7 6
 4 3 2 1 12 11 10 9 8 7 6 5

(a) `int i = n * n;`

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

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

`System.out.print(c -- + " ");`

`} // inner loop`

`}`

`System.out.println();`

`}`

`}`

65. 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 0 0 0 0 0

1 1 1 1 1

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

0 0 0 0 0

`if (i % 2 == 0)`

1 1 1 1 1

`System.out.print(i);`

`} // inner loop`

`if (i <= n - 1)`

`System.out.print(",");`

`}`

`System.out.println();`

`}`

	j=1	2	3	4	5
i=1	5	4	3	2	1
i=2	10	9	8	7	6
i=3	15	14	13	12	11
i=4	20	19	18	17	16
i=5	25	24	23	22	21

 $n = 5$ logic: $n * i$

{Q1}

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

int c = n * i;

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

System.out.print(c-- + " ");

System.out.println();

}

67. 1 1 1 1 1

0 0 0 0 0

1 1 1 1 1

0 0 0 0 0

1 1 1 1 1

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

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

System.out.print((i%2 + j%2) + " ");

System.out.println();

}

68. 1 0 1 0 1
 1 0 1 0 1
 1 0 1 0 1
 1 0 1 0 1
 1 0 1 0 1

\Rightarrow for (int i=1; i<=n; i++) {
 for (int j=1; j<=n; j++) {
 System.out.print(j%2 + "");}
 System.out.println();
 } $\{$ $\}$ $\{$ $\}$

69. 0 1 0 1 0
 1 0 1 0 1
 0 1 0 1 0
 1 0 1 0 1
 0 1 0 1 0

\Rightarrow for (int i=0; i<=n; i++) {
 for (int j=1; j<=n; j++) {
 System.out.print((i+j)%2);}
 System.out.println();
 } $\{$ $\}$ $\{$ $\}$

70.

1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	

int c=1;
~~for (int i=0; i<n; i++)~~

{
~~for (int j=0; j<=i; j++)~~

System.out.print(c);

c++;

} System.out.println();

3

71.

0	1	0	1	0	
1	0	1	0	1	
2 * 3	0	1	0	1	0
4 * 5 * 6	1	0	1	0	1
7 * 8 * 9 * 10	0	1	0	1	0

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

{
~~for (int j=0; j<=i; j++)~~

System.out.print(c);

c++;

} if (j < i)

System.out.print("*");

} System.out.println();

b)

while loopinit $i = 0$

while (condition)

{

 $i++;$

{

Do while loop

Do

{

init $i = 0;$;
 ;

{

while (condition);
 $i++;$ For loopfor (initialization,
condition,
update)

{

 ;

{

j,	1	2	3	4	5	
12	15	*	3	*	*	$n = 5$
14		13				
3	12	11	10			$c = \frac{n(n+1)}{2}$
4	9	8	7	6		
5	5	4	3	2	1	

mt $c = n * (n+1) / 2$ In $n = 15 \Rightarrow 15$ $n = 4 \Rightarrow 10$ $n = 3 \Rightarrow 6$ for (mt $i = 1; i < n; i++$)

{

for (mt $j = 1; j < n; j++$)

{

System.out.print(c);

c--;

{

System.out.println();

{

10

 $n = 4$ 9 8

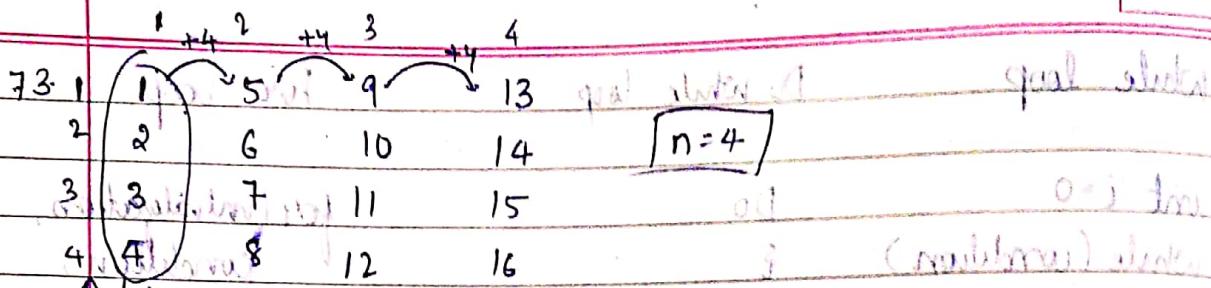
7 6 5

4 3 2 1

 $n = 3$ 6

5 4

3 2 1



Q2 Pattern

geo

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

System.out.println(c);

c=c+n;

}

System.out.println();

}

74.

1	2	3	4	5	6
1	1	1	1	2	3
2	2	2	2	3	4
3	3	3	3	4	5
4	4	4	4	5	6
5	5	5	5	6	7

c=1

j<n

Sol

75.

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

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

, 1 2 3 4

(x-1)	1	1	1	1	2
2	3	2	2	2	2
3	3	3	3	3	4
4	5	4	5	4	4

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

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

{
if (i%2 == 0) {
System.out.print(i+1);
}
}

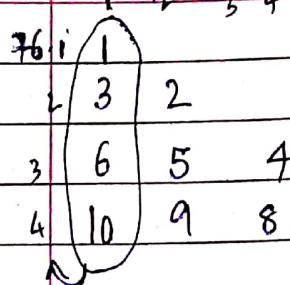
for (int j=1; j<=n; j++)

{
System.out.print(j);
}

if (i%2 != 0)
{

System.out.print(i+1);
}
}

System.out.println();
}



for n=4

for n=3

1
3 2

6 5 4

$$i*(i+1)/2$$

int c = 1;

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

$$c = i * (i+1) / 2;$$

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

{

 System.out.print(c);

 c--;

}

 System.out.println();

}

0 1 2 3 4
 77. ~~1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16~~
 26 * 27 * 28 * 29 * 30
 22 * 23 * 24 * 25
 19 * 20 * 21
 17 * 18

C-1

Sol. $c+j$ 1 2 3 4 5
 $c-2 + 1^3$

int $c=1$; int $c1=n*h+1$; ~~int~~ int $k=n-1$;
 \Rightarrow for (int $i=0$; $i < n$; $i++$)

for (int $j=0$; $j \leq i-1$; $j++$)

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

for (int $j=0$; $j < n-i$; $j++$)

System.out.print (c);

$c++$;

System.out.print ("*");

$\}$.

for (int $j=0$; $j < n-i$; $j++$)

System.out.print ($c1$);

System.out.print ("*");

$\}$.

~~process~~ $c1 = c1 - k$;

$k--$;

System.out.println();

$$\begin{aligned}
 & \text{Q) } 36 \xrightarrow{\text{1}} 6 \times 6 \\
 & \quad \downarrow \text{①} \\
 & 36 \rightarrow 6 \times \textcircled{6} \\
 & \quad \downarrow \text{②} \\
 & 6 \rightarrow 2 * \textcircled{3} \\
 & \quad \downarrow \text{③} \\
 & 3 \Rightarrow \textcircled{2} + 1 \\
 & \quad \downarrow \text{④}
 \end{aligned}$$

①) class Try {

PSVM ()

§

$$S.O.P \quad (0.1 + 0.1 + 0.1 = 0.3),$$

$$S.O.P \quad (0.1 + 0.1 + 0.1 = 0.3);$$

O/P: false . Because $0.1 + 0.1 + 0.1 = 0.300000004$
and $0.3 \neq 030000004$

But if we type cast it it will give true - because of me

0) 0 1 2 3 4 5

1 11 12 13 14 15

2 21 22 23 24 25

-3) 16 17 18 19 20

4 6 7 8 9 10

$n=5$

$$n=5 \quad (n+i)+1$$

$$A = n * (i+1) + 1 \quad 5 * 3 + 1$$

$$(5 * 2) + 1 \Rightarrow 11.$$

$$(5 * 4) + 1 \Rightarrow 21$$

$$5 * (2+1) + 1$$

$$5 * 3 + 1$$

$$16.$$

5 11 12 13 14 15

6 17 18 19 20

7 21 22 23 24 25

8 26 27 28 29 30

9 31 32 33 34 35

10 19 20 21 22 23

11 7 8 9 10 11

$n=6$

Q2a

$c=1$

1 13 14 15 16 17 18

2 25 26 27 28 29 30

3 31 32 33 34 35 36

4 19 20 21 22 23 24

5 7 8 9 10 11 12

$$1 \rightarrow 2 \rightarrow c+2n$$

$$2 \rightarrow 4$$

$$3 \rightarrow 5$$

$$4 \rightarrow 3$$

$$5 \rightarrow 1$$

(classmate) for 20+10 = 30

$n=5$

0	1	2	3	4	5	$c + n$
$c + 2n$	11	12	13	14	15	$c + 2n$
$c + 2n$	21	22	23	24	25	$c + 4n$
$c + 3n$	16	17	18	19	20	$c + 3n$
$c + 4n$	6	7	8	9	10	$c + n$

$n=4$

1	2	3	4	$c + 2n$
$c + 2n$	9	10	11	12
$c + n$	13	14	15	16
$c - 2n$	5	6	7	8

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

{

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

 {

 System.out.print(c + j + " ");

 }

 if (n % 2 == 0 || i == (n / 2) - 1)

 c = c + n;

}

 else if (n % 2 != 0 || i != n / 2)

 c = c - n;

}

 else if (i >= n / 2)

 c = c - 2 * n;

}

 else

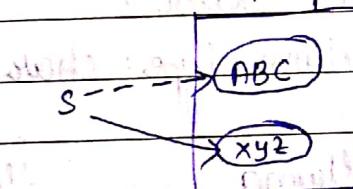
$c = c + s * n;$

`System.out.println();`

* STRINGS

- String is a class.
- It can accept all kind of characters (alphabets, numbers, special characters etc).
- Immutable String: A string which cannot be changed.

e.g: `String s = "PBC";`
`s = "XYZ";`



`System.out.println(s);`

o/p: XYZ.

Strings: `String s = "ABC";`

e.g. `s = "123";`

`s = " ";`

`s = "AB12";`

`s = " ";` //Empty String

`s = "A";`

`s = "$@ - !";`

- Methods () are bounded to class whereas as functions () are not bounded to class, even if the class is not present we can use functions but we cannot use methods without a class.

* String Methods:

1. length():

→ Return type: int

Eg: String s = "Hello Programmers";

System.out.print(s.length());

O/P: 14

2. charAt(index)

→ Return type: char

Eg: String s = "Hello";

System.out.print(charAt(2));

O/P: L

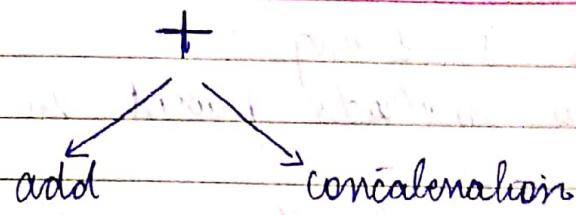
3. toCharArray()

→ Return type: String type array.

Eg: String s = "Hello";

char[] ar = s.toCharArray();

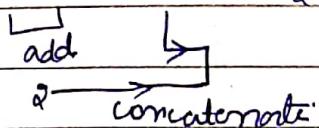
O/P: ar → [H|e|l|l|o]



① $1 + 1 + 1 = 3$ (All are integers, they will be added).

② ' $'1'$ ' + 1 $\Rightarrow 50$ (' $'1'$ ' is character so computer takes its ASCII value i.e., 49 so $49 + 1 = 50$)

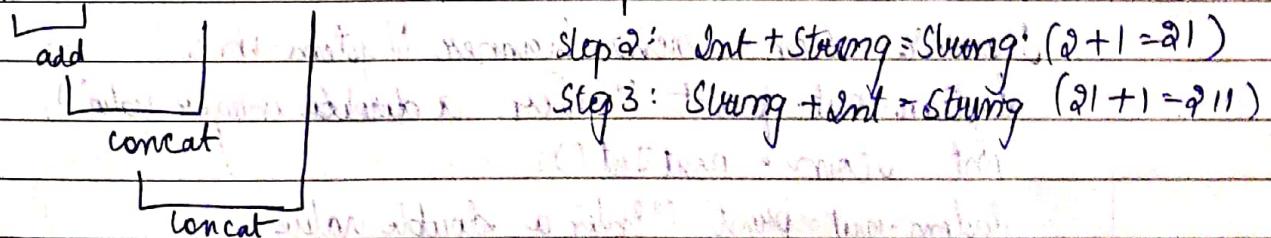
③ $1 + 1 + '1' \Rightarrow 2 + 49 = 51$



④ $1 + '1' + 1 = 1 + 49 + 1 = 51$ (Both are strings)

⑤ " $'1'$ " + 1 + 1 $\Rightarrow 111$ Step 1: String + integer = String (11)
String + integer = String (111).

⑥ 1 + 1 + " $'1'$ " + 1 $\Rightarrow 211$ Step 1: Int + Int = Int ($1 + 1 = 2$)



* How to take I/P strings?

A: We use two methods present in Scanner class

① next() - It does not allow spaces

Eg: ABCforTech.

② nextLine() - It allows spaces.

Eg: ABC for Tech.

Scanner

↓
next() nextLine()

* WAP to take an integer input then to take double as input and then to take a string as I/P then print them.

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

(or) `class DataInput`

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

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

```
System.out.print ("Enter a double integer value");
```

```
int x=scan.nextInt();
```

```
System.out.print ("Enter a double value");
```

```
double d=scan.nextDouble();
```

```
scan.nextLine();
```

```
System.out.print ("Enter a String value");
```

```
String s= scan.nextLine();
```

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

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

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

}

Q) WAP to print ASCII value of any given I/P character.

Sol: import java.util.Scanner;

class ASCII

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter a char");

char c = scan.next ().charAt (0);

System.out.print ("The value is :" + (int)c);

}

Ans: ASCII Values: a - z : 97 - 122, special ch - 32 - 47,
58 - 64

A - Z : 65 - 90, 91 - 96

108 - 127

Q) WAP to count the spaces in string.

Sol: class CountSpace

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter a string");

String s = scan.nextLine ();

int count = 0;

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

{

if (s.charAt (i) == ' ')

{

```
count++;
```

{

{

```
System.out.print(count);
```

{

{

I/O: This is my class.

O/P: 3

H/W: 1. Write a program to reverse a string

2. Write a program to print all lower case

alphabets

3. Write a program to calculate area of rectangle

* Pin all the string programs written in aplink book.

a) WAP to perform swapping of two numbers using 3rd variable

```
Sol> import java.util.Scanner;
class Swap
{

```

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

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

```
        int x = scan.nextInt();
```

```
        int y = scan.nextInt();
```

```
        System.out.print ("Before swapping x = " + x + " y = " + y);
        temp = x; // swap
```

```
        x = y;
```

```
        y = temp;
```

```
        System.out.print ("After swapping x = " + x + " y = " + y);
```

b) WAP to sort the given string.

```
Sol> import java.util.Scanner;
```

```
class Sorting
{

```

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

```

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

```
        System.out.print ("Enter a String");
```

```
        String str = scan.nextLine();
```

```
        char arr [] = str.toCharArray ();
```

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

```

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

```

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

```

char t = a[i];

a[i] = a[j];

a[j] = t;

}

}

}

}

}

}

g/p: acbaaccb

If we use '*<*' in

if condition we

will get the array in
increasing sorted order

and if we use '*>*'

it will be in decreasing
sorted order.

o/p: aaabbccc

Q) WAP to remove all special characters and
numbers from a given input string

Q) g/p: — ab\$*bc*d123 at 123 at 123 (e
abbcd.

import java.util.Scanner;

class RemoveSpecialChar { public static void main(String args[]) {

Scanner scan = new Scanner(System.in);

System.out.print("Enter a string:");

String s = scan.nextLine();

String res = " ";

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

if ((s.charAt(i) >= 65 && s.charAt(i) <= 90) ||

s.charAt(i) >= 97 && s.charAt(i) <= 122)

System.out.print(temp);

- Q) WAP to place all the special characters present in the string to the end of the string.

I/p: ABC@for tech@#

O/p: ABCfor tech@#@#

```
import java.util.Scanner;
class End {
    public static void main() {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter a String");
        String s = scan.nextLine();
        String res = "";
        String temp = "";
        for (int i = 0; i < s.length(); i++) {
            if (!((s.charAt(i) >= 65 & s.charAt(i) <= 90) ||
                  s.charAt(i) >= 97 & s.charAt(i) <= 122) ||
                  s.charAt(i) >= 48 & s.charAt(i) <= 57) {
                res += s.charAt(i);
            } else {
                temp += s.charAt(i);
            }
        }
        System.out.print(res + temp);
    }
}
```

```
if (!((s.charAt(i) >= 65 & s.charAt(i) <= 90) ||
      s.charAt(i) >= 97 & s.charAt(i) <= 122) ||
      s.charAt(i) >= 48 & s.charAt(i) <= 57) {
    res += s.charAt(i);
}
```

temp += s.charAt(i);

```
if ((s.charAt(i) >= 65 & s.charAt(i) <= 90) ||
    s.charAt(i) >= 97 & s.charAt(i) <= 122) ||
    s.charAt(i) >= 48 & s.charAt(i) <= 57) {
    res += s.charAt(i);
}
System.out.print(res + temp);
```

Q) Write a program to remove a specific character from the given string.

\hookrightarrow import java.util.Scanner;
class RemoveSPC {

public static void main(String args[]) {

Scanner scan = new Scanner(System.in);

System.out.print("Enter a string");

String s = scan.nextLine();

System.out.print("Enter a character to be removed");

Char k = scan.next().charAt(0);

String res = " ";

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

if (s.charAt(i) != k) {

res += s.charAt(i);

System.out.println(res);

System.out.println("The original string is " + s);

O/P: ABCD for TECH. st = <(0)->ABCD

O/P: ABC for TECH. st = <(0)->ABC

Q)

1

$$2 * 3$$

$$4 * 5 * 11 * 6$$

$$7 * 8 * 9 * 10$$

$$7 * 8 * 9 * 10$$

$$4 * 5 * 6$$

$$2 * 3$$

1

Sol \Rightarrow operation between all pair of $(2, 3, 5)$ will be

$$= [1 - (3 \cdot 5 \cdot 2 + 3 \cdot 5) - 1] + 20 = \text{answer}$$

$$f = k \text{ per min} \quad \frac{k}{3} \text{ per sec}$$

$$[1 - (3 \cdot 5 - 3 \cdot 5) - 1] + 20$$

$$(1 - 1) + 20$$

$$20 + 0$$

$$20$$

\therefore an answer of 20

26/11/2019

* Encryption Or Encoding

A	B	C	D	E	F	G	H	I	J
65	66	67	68	69	70	71	72	73	74
K	L	M	N	O	P	Q	R	S	T
75	76	77	78	79	80	81	82	83	84
U	V	W	X	Y	Z		S	T	
85	86	87	88	89	90				

Upper Case: for calculating the encrypted message.

Formula: $65 + [k - (90 - s \cdot \text{charAt}(i) - 1)]$

Eg: Encrypt Z, and key k = 4

$$65 + [4 - (90 - 90) - 1]$$

$$65 + (4 - 1)$$

$$65 + 3$$

$$68.$$

So Z will be encrypted as D.

Encrypting upper case: Input: abcde Output: x,y A key is
o/p: B,C E

import java.util.Scanner;
class Encryption

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

Scanner scan = new Scanner (System.in);

System.out.print ("Enter a string");

String s = scan.nextLine();

System.out.print ("Enter a key");

int k = scan.nextInt();

String mes = ".(initial message)";

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

{

if (s.charAt(i)>=65 && s.charAt(i)<=90 ||

s.charAt(i)>=97 && s.charAt(i)<=122)

{

Upper Case if (s.charAt(i)>=65 && s.charAt(i)<=90 - K), checking Range

else if (s.charAt(i)>=97 && s.charAt(i)<=122 - K), checking Range

res += (char) (s.charAt(i)+K);

else res += (char) (65 + (K - (90 - s.charAt(i))));

else if (s.charAt(i)>=97 && s.charAt(i)<=122 - K)

res += (char) (s.charAt(i)+K);

else

res += (char) (97 + (K - (122 - s.charAt(i))));

→ 3

TOPIC : STRING

else if (s.charAt(i) >= 48 && s.charAt(i) <= 57) .
 ↳ checking range of no's

if (s.charAt(i) >= 48 && s.charAt(i) <= 57 - k);
 ↳ checking if pos i is not greater than k

res += s.charAt(i) + k;
 ↳ (English script) norm. ↳ required as we have print only.
 Here no type casting is required as we have print only.

else.
 ↳ Criterions satisfied with respect to print
 res += 48 + (k - (57 - s.charAt(i)) - 1);
 ↳ minimum value is reached

↳ also if closed (if pos in print) found transformation
 ↳ for loop closed

~~seperates~~ System.out.print(res);
 ↳ output displayed on console

↳

Q. Ques = 2/11 How many file exist in C:\Windows\ ?
 (See) SUBSTRINGS = part of a string

1. contains (): It will check whether a given character is present in the string.
2. SubString (): It will split the string at given inputs.

Eq: s = tli my name

if ((i + m) - o <= 2) + 1) (m) - 1) = 7 8 9

1. SubString (" ");

2. SubString (0,5)

(i + l) - s1 = 1 + 1 - 2 = 0 & FP = < (j) s1 = tli ; my

S2 = my

S2 = name

S = name , (x + (j) JA. m) - 2) (m) - 1) = 7 8 9

if ((i + (j) JA. m) - 2 - s1) - 1) + FP) (m) - 1) = 7 8 9

Q) WAP to check whether a substring is present in string or not.

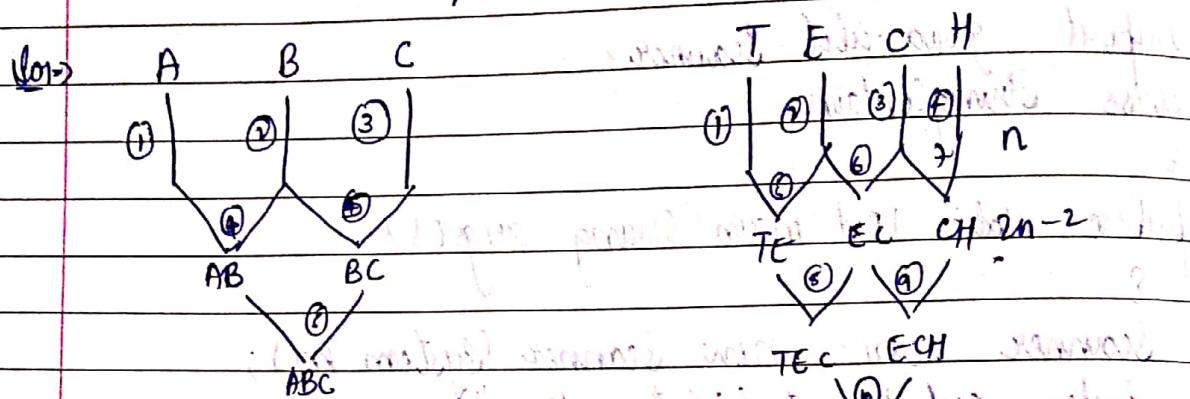
```

import java.util.Scanner;
class StringContain {
    public static void main(String args[]) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter string 1");
        String s1 = scan.nextLine();
        System.out.print("Enter string 2");
        String s2 = scan.nextLine();
        boolean flag = false;
        for (int i = 0; i < s1.length(); i++) {
            if (s1.contains(s2))
                flag = true;
        }
        if (flag == true)
            System.out.print("yes");
        else
            System.out.print("no");
    }
}
  
```

I/P: s1: Hyderabad s2: f and

O/P: no

Q) WAP to find the number of substrings present inside a string. (here substring has no spaces)



for substring (without spaces) formula $\frac{n * n + 1}{2}$

" " " with " $\frac{n * n + 1}{2}$ " for

```
import java.util.Scanner;
class CountSubstrings
```

```
{
```

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

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

```
System.out.print("Enter a string");
```

```
String s = scan.nextLine();
```

```
int n = s.length();
```

```
int res = n * (n + 1) / 2;
```

```
System.out.print(res);
```

```
}
```

I/P: ABC

O/P: 6

a) NAP to insert a substring in between a string

~~import java.util.Scanner;~~

class InsertSubString

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Input original String");

String s1 = scan.nextLine();

System.out.print ("Input the string to be inserted");

String s2 = scan.nextLine();

System.out.print ("Enter an index value");

int n = scan.nextInt();

String res = "";

for (int i=0; i < s1.length(); i++)

{

rest = s.charAt(i);

if (i == n)

rest = s2;

}

System.out.print (res);

}

I/O: s1 = abc tech

s2 = for

n = 3

O/P: res = abcfor tech.

30/7/2019

Q) NAP to reverse ~~a~~ multiple string
GP: how are you.
O/P: woh era uoy

Scanner scan = new Scanner(System.in);

System.out.print("Enter the string");

String s = scan.nextLine();

int count = 0;

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

{ if (s.length(i) == ' ' && s.length(i+1) == ' ')

s.o.p(count++);

}

}

String ar[] = new String(count++);

for (int i = ar.length; i >= 0; i--)

{ if (ar.length! == ' ')

s.o.p(ar[i]);

}

else if (ar.length == ' ')

ar[j] = t;

t = ' ';

}

ar[i]++;

}

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

{ s.o.p(ar[i]);

}

3

* Complexity Of Algorithms

Complexity (Time)

algorithm (Space)

less time

less space

Complexity is the time taken to solve a problem.

- In order for an algorithm to be good it should take less time and space.

* Time complexities:

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

$\{$
} n times these will execute

Time complexity: $O(n)$

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

$\{$
 $n \{$
 $n \{$
 $n \{$
} } } }

Time complexity: $O(n * n)$
 $O(n^2)$

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

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

i	j
0	0
1	0, 1
2	0, 1, 2

Time complexity: $O(n * (n+1))$

$O(n^2 + n)$ We can ignore lower power.

$O(n^2)$

* Space complexity: A program taking more space.

① Variables → ① Dynamic variable

② Static variable

② Environment (program)

- Merge sort has highest time complexity.

ARRAY

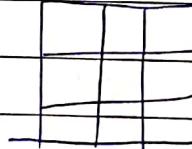
- Array is a collection of homogeneous type of data
- There are three types of arrays

① 1D array: It will take only one dimension

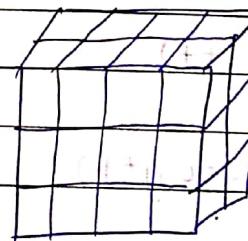
Eg.

1	2	3
---	---	---

② 2D array:



③ Multi D array:



* 1 Dimensional Array:

- To enter the elements and print the elements.

```

import java.util.Scanner;
class Arr
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the size of array");
        int n = scan.nextInt();
        int arr[] = new int[n];
        for (int i = 0; i < n; i++)
        {
            System.out.print("Enter the element at: " + i);
            arr[i] = scan.nextInt();
        }
        for (int j = 0; j < arr.length; j++)
        {
            System.out.print("The elements are: " + arr[j]);
        }
    }
}
  
```

Q) WAP to add all elements of the array.

```
Sol: import java.util.Scanner;  
class SumArray  
{  
    public static void main (String args[])  
    {  
        Scanner scan = new Scanner (System.in);  
        System.out.println ("Enter the size of array");  
        int n = scan.nextInt();  
        int res = 0;  
        int a[] = new int [n];  
        for (int i = 0; i < n; i++)  
        {  
            System.out.println ("Enter the element at " + i);  
            a[i] = scan.nextInt();  
        }  
        for (int i = 0; i < a.length; i++)  
        {  
            res = res + a[i];  
        }  
        System.out.println ("The sum is: " + res);  
    }  
}
```

I/P: n = 3
4, 5, 1

O/P: 10

3) WAP to add the elements of an array:

import java.util.Scanner;

class AddArray

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the size of array");

int n = scan.nextInt();

int res = 0;

int a[] = new int [n];

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

{

System.out.print ("Enter the element at: " + i)

a[i] = scan.nextInt();

}

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

{

if (i == 0) q = 0

{

res = res + a[i];

}

}

System.out.print (res);

}

}

O/P: enter the size of array: 4

2 3 4 5

0 1 2 3

O/P: 6.

5) WAP to add the elements present at odd index.

Ans import java.util.Scanner;

class Sum OddIndexSum

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.println ("Enter the size of array");

int n = scan.nextInt();

int a[] = new int [n];

int res = 0;

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

{

System.out.print ("Enter the element :" + i);

a[i] = scan.nextInt();

{

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

{

else if (i % 2 != 0)

{

res += a[i];

{

{

System.out.print ("The sum is :" + res);

{

{

S/P: n = 3

4 5 6
0 1 2

O/P: 5.

Q) WAP to multiply the elements of given array.

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

```
class ProductOfArray
```

```
{
```

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

```
{
```

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

```
System.out.println ("Enter the size of array");
```

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

```
int a [] = new int [n];
```

```
int res = 1;
```

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

```
{
```

```
System.out.print ("Enter the element : " + i);
```

```
a [i] = scan.nextInt();
```

```
}
```

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

```
{
```

```
res = res * a [i];
```

```
}
```

```
System.out.print (res);
```

```
}
```

(2nd attempt - 1st attempt)

(2nd attempt - 2nd attempt)

(2nd attempt - 3rd attempt)

(2nd attempt - 4th attempt)

(2nd attempt - 5th attempt)

(2nd attempt - 6th attempt)

(2nd attempt - 7th attempt)

(2nd attempt - 8th attempt)

(2nd attempt - 9th attempt)

(2nd attempt - 10th attempt)

(2nd attempt - 11th attempt)

(2nd attempt - 12th attempt)

(2nd attempt - 13th attempt)

(2nd attempt - 14th attempt)

(2nd attempt - 15th attempt)

(2nd attempt - 16th attempt)

(2nd attempt - 17th attempt)

(2nd attempt - 18th attempt)

(2nd attempt - 19th attempt)

(2nd attempt - 20th attempt)

(2nd attempt - 21st attempt)

(2nd attempt - 22nd attempt)

(2nd attempt - 23rd attempt)

(2nd attempt - 24th attempt)

(2nd attempt - 25th attempt)

(2nd attempt - 26th attempt)

(2nd attempt - 27th attempt)

(2nd attempt - 28th attempt)

(2nd attempt - 29th attempt)

(2nd attempt - 30th attempt)

(2nd attempt - 31st attempt)

(2nd attempt - 32nd attempt)

(2nd attempt - 33rd attempt)

(2nd attempt - 34th attempt)

(2nd attempt - 35th attempt)

(2nd attempt - 36th attempt)

(2nd attempt - 37th attempt)

(2nd attempt - 38th attempt)

(2nd attempt - 39th attempt)

(2nd attempt - 40th attempt)

(2nd attempt - 41st attempt)

(2nd attempt - 42nd attempt)

(2nd attempt - 43rd attempt)

(2nd attempt - 44th attempt)

(2nd attempt - 45th attempt)

(2nd attempt - 46th attempt)

(2nd attempt - 47th attempt)

(2nd attempt - 48th attempt)

(2nd attempt - 49th attempt)

(2nd attempt - 50th attempt)

(2nd attempt - 51st attempt)

(2nd attempt - 52nd attempt)

(2nd attempt - 53rd attempt)

(2nd attempt - 54th attempt)

(2nd attempt - 55th attempt)

(2nd attempt - 56th attempt)

(2nd attempt - 57th attempt)

(2nd attempt - 58th attempt)

(2nd attempt - 59th attempt)

(2nd attempt - 60th attempt)

(2nd attempt - 61st attempt)

(2nd attempt - 62nd attempt)

(2nd attempt - 63rd attempt)

(2nd attempt - 64th attempt)

(2nd attempt - 65th attempt)

(2nd attempt - 66th attempt)

(2nd attempt - 67th attempt)

(2nd attempt - 68th attempt)

(2nd attempt - 69th attempt)

(2nd attempt - 70th attempt)

(2nd attempt - 71st attempt)

(2nd attempt - 72nd attempt)

(2nd attempt - 73rd attempt)

(2nd attempt - 74th attempt)

(2nd attempt - 75th attempt)

(2nd attempt - 76th attempt)

(2nd attempt - 77th attempt)

(2nd attempt - 78th attempt)

(2nd attempt - 79th attempt)

(2nd attempt - 80th attempt)

(2nd attempt - 81st attempt)

(2nd attempt - 82nd attempt)

(2nd attempt - 83rd attempt)

(2nd attempt - 84th attempt)

(2nd attempt - 85th attempt)

(2nd attempt - 86th attempt)

(2nd attempt - 87th attempt)

(2nd attempt - 88th attempt)

(2nd attempt - 89th attempt)

(2nd attempt - 90th attempt)

(2nd attempt - 91st attempt)

(2nd attempt - 92nd attempt)

(2nd attempt - 93rd attempt)

(2nd attempt - 94th attempt)

(2nd attempt - 95th attempt)

(2nd attempt - 96th attempt)

(2nd attempt - 97th attempt)

(2nd attempt - 98th attempt)

(2nd attempt - 99th attempt)

(2nd attempt - 100th attempt)

O/P: n = 3

1 2 3

O/P: 6

Scanned with CamScanner

Q) WAP to multiply the array elements at odd index

```
import java.util.Scanner;  
class ProductOddIndex  
{  
    public static void main(String args)  
    {  
        Scanner scan = new Scanner(System.in);  
        System.out.print("Enter the size of array");  
        int n = scan.nextInt();  
        int res = 1;  
        int arr[] = new int[n];  
        System.out.print("Enter the elements");  
        for (int i = 0; i < n; i++)  
        {  
            arr[i] = scan.nextInt();  
        }  
        for (int i = 0; i < arr.length; i++)  
        {  
            if (i % 2 != 0)  
            {  
                res = res * arr[i];  
            }  
        }  
        System.out.println(res);  
    }  
}
```

Q) NAP to find the sum of the elements of an array.

Ans: import java.util.Scanner;

class Mean

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the size of array");

int n = scan.nextInt();

int sum = 0;

int float avg = 0;

int arr[] = new arr[n];

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

{

System.out.print ("Enter the element: " + i);

arr[i] = scan.nextInt();

}

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

{

temp = temp + arr[i];

}

avg = (float) temp / n;

System.out.print (avg);

}

}

Sol: n = 5

4 6 5 7 3

O/p: 5.0

(8) WAP to search the particular element in array present at a particular index.

Sol: import java.util.Scanner;

class SearchElement

{

public static void main (String args[])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the size of an array");

int n = scan.nextInt();

int ar[] = new int [n];

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

{

System.out.print ("Enter the element "+i);

ar[i] = scan.nextInt();

}

System.out.print ("Enter the key value");

int k = scan.nextInt();

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

{

if (i == k)

{

System.out.print ("The element is "+ar[i]);

}

}

}

Q/P: n = 5

1 4 8 9 6

K = 3

The element is ③.

31/3/2019

a) find the second largest number in an array.

import java.util.Scanner;

class SecondLargestNum

{

public static void main (String args [])

{

Scanner scan = new Scanner (System.in);

System.out.print ("Enter the size of Array");

int n = scan.nextInt();

int ar [] = new int [n];

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

{

System.out.print ("Enter the element: " + i);

ar[i] = scan.nextInt();

{

int fl = 0;

int sl = 0;

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

{

sl = fl;

fl = ar[i];

else if (ar[i] > sl)

{

sl = ar[i];

{

{

S.O.P (sl);

{

{

* To check duplicates and ignore them from previous program.

import java.util.Scanner;

else if ($a[i] > sl$ && $f1 \neq a[i]$)

{

$sl = a[i];$

}

S.O.P(sl);

}

}

* To find the second largest no. in -ve numbers.

import java.util.Scanner;
class NegSecondLargestNum

public static void main (String args[])

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

System.out.print ("Enter the size of array");

int n = scan.nextInt();

int ar [] = new int [n];

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

{

System.out.print ("Enter the element "+i);

$a[i] = scan.nextInt();$

}

int f1=0;

int sl=0;

if ($a[0] > a[i]$)
 {

$fl = a[0];$

$sl = a[0];$

}

else {

$fl = a[i];$

$sl = a[0];$

}

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

if ($a[i] > fl$)

$sl = fl;$

$fl = a[i];$

}

else if ($a[i] > sl \& fl != a[i]$)

{

$sl = a[i];$

}

}

System.out.println(sl);

{

}

O/p: $n = -5$

-1 -4 -2 -8 -6

O/p: -2

* I/P: 3 3 3 2 1

O/P: 2

$$1 * 2 * 3 * 4 * 5 = 120$$

$$\underline{1 * 2 * 3 * 4 * 5} = 120 / 2 = 60$$

* I/P: [1 9 3 4 5]

$$\underline{1 * 2 * 3 * 4 * 5} = 120 / 3 = 40$$

O/P:

[120 60 40 30 24]

$$\underline{1 * 2 * 3 * 4 * 5} = 120 / 4 = 30$$

$$\underline{1 * 2 * 3 * 4 * 5} = 120 / 5 = 24$$

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

{

$$\text{res}[i] = \underline{a[0][i+1]} * \underline{a[0][i+2]} * \underline{a[0][i+3]} * \underline{a[0][i+4]};$$

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

{

Q1) `import java.util.Scanner;`

`class ProductOfArray`
{

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

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

`System.out.print("Enter the size:");`

`int n = scan.nextInt();`

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

`System.out.print("Enter the elements");`

`a[i] = scan.nextInt();`

}

`int res[] = new int[n];`

`int p=1;`

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

`p = p * a[i];` $\rightarrow 120$

}

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

 Σ

$arr[i] = \text{Play}(i) \rightarrow m_1 = 120, m_2 = 60, m_3 = 40, m_4 = 30, m_5 = 24$

System.out.print("res(" + " ");

 Σ
 Σ
 Σ

Step: $n = 5$

arr [1|2|3|4|5]

Op: [120|60|40|30|24]

Q) Op: [1|2|3|4|5]

$n = 5$

Op: [14 10].

$$2 + 3 + 4 + 5 = 14$$

$$1 + 3 + 4 + 5 = 13$$

$$1 + 2 + 4 + 5 = 12$$

$$1 + 2 + 3 + 5 = 11$$

$$1 + 2 + 3 + 4 = 10$$

$$15 - 1 = 14$$

$$15 - 5 = 10$$

(*) WAP to find the largest sum and the smallest sum in n array.

Let's import java.util.Scanner;
class MinMax
{

public static void main (String args[])

Scanner scan = new Scanner (System.in);
System.out.print ("Enter the size");

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

```
{  
    System.out.print("Enter the element");
    arr[i] = scan.nextInt();
}
```

```
int sum = 0;
int min = arr[0], max = arr[0];
for (int i = 0; i < arr.length; i++)
```

```
{  
    sum = sum + arr[i];
}
```

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

```
    if (arr[i] > max)
    {
```

```
        max = arr[i];
    }
```

```
    else if (arr[i] < min)
    {
```

```
        min = arr[i];
    }
```

```
}  
System.out.print((sum - min) + " " + (sum - max));
```

```
}
```

* Recursion: Recursion is nothing but a function calling itself repeatedly.

Factorial Recursion program

Main()

{

System.out.print (factorial)

;

static int fact (int n)

{

int fact;

if (n == 0)

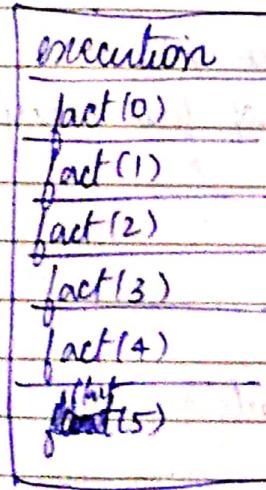
return 1;

else

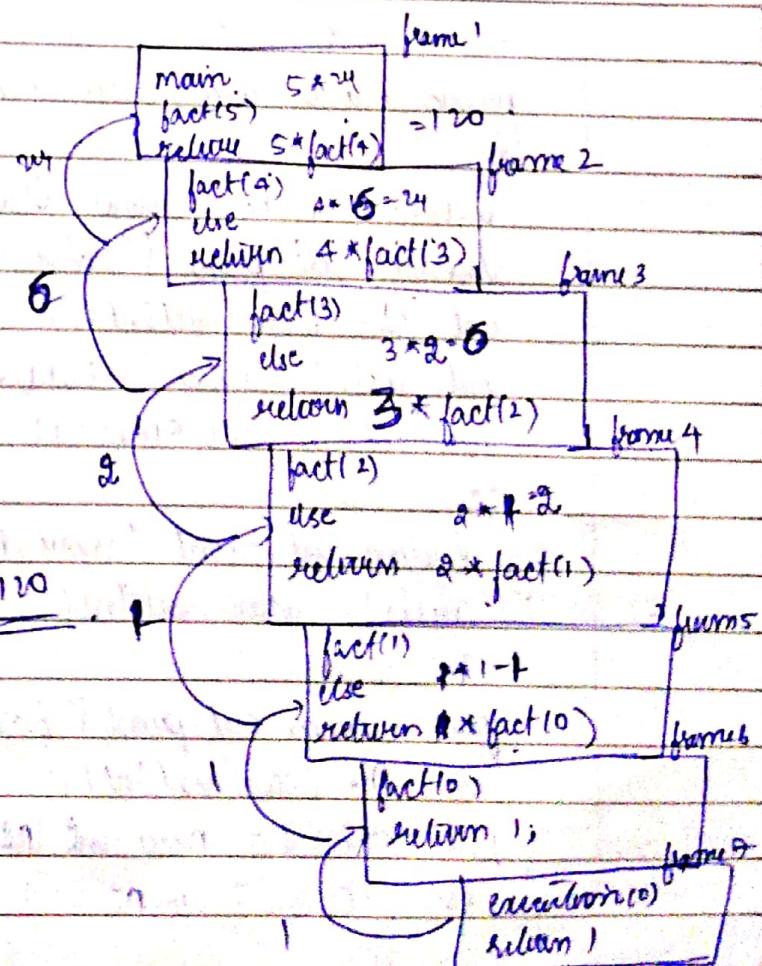
fact = n * fact(n-1);

}

Stack:



$$\underline{\text{factorial} = 120}$$



zig zag

- Q) WAP to insert an array in its ~~exact~~ format!
Merge the array in its ~~exact~~ format.

→ arr1

1	3	5
---	---	---

arr2

2	4	6	7	8
---	---	---	---	---

O/p: res

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

(a) arr1

4	5	6
---	---	---

arr2

1	2	3	4
---	---	---	---

O/p: res

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

→ import java.util.Scanner;
 ↳ includes MergeArray
 {

public static void main(String args[])
 {

Scanner scan = new Scanner(System.in);

System.out.print("Enter the size of first array");

int n1 = scan.nextInt();

int arr1[] = new int[n1];

for (int i = 0; i < n1; i++)
 {

System.out.print("Enter the elements");

arr1[i] = scan.nextInt();

for (int i = 0; i < n1; i++)
 System.out.print(arr1[i]);

int n2 = scan.nextInt();

int arr2[] = new int[n2];

for (int i = 0; i < n2; i++)
 {

{

System.out.print("Enter the elements.");
arr[0] = scan.nextInt();

}

int res = max - arr[0];

int max = 0;

if (arr1.length > arr2.length)

 max = arr1.length;

}

else

 max = arr2.length;

int j = 0; // for placing in the max size array

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

 {

 if (i < arr1.length)

 {

 res[j] = arr1[i];

 j++;

 }

 if (i < arr2.length)

 {

 res[j] = arr2[i];

 j++;

 }

}

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

 {

 System.out.print(res[i]);

 }

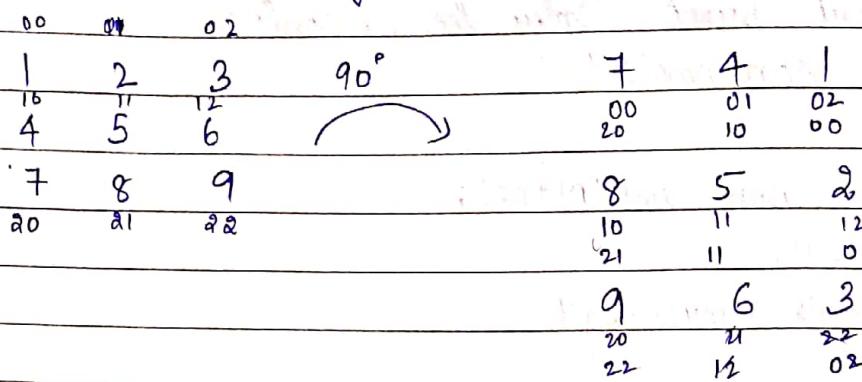
}

} // for loop for printing the resultant array

} // for loop for printing the resultant array

} // for loop for printing the resultant array

Q) Rotate a 2D array 90°.



```
import java.util.Scanner;
class Rotate90
{
```

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

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

```
System.out.print ("Enter the size");
```

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

```
int arr[][] = new int [n][n];
```

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

```
{
```

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

```
{
```

```
System.out.print ("Enter the elements");
```

```
arr[i][j] = scan.nextInt();
```

```
}
```

```
} int res[][] = new int [n][n];
```

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

```
{
```

int k = n - 1;

```
for (int j = 0; j < arr.length; j++, k--)
```

```
{
```

```
res[i][j] = arr[k][i];
```

```
{
```

System.out.println ("Result");

Scanned with CamScanner

```
for (int i=0; i<res length; i++)
{
```

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

```
        System.out.print (res[i][j]);
    }
```

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

{

{

* class A

{

```
static void print()
```

{

```
System.out.print ("June 10");
}
```

{

class ABC

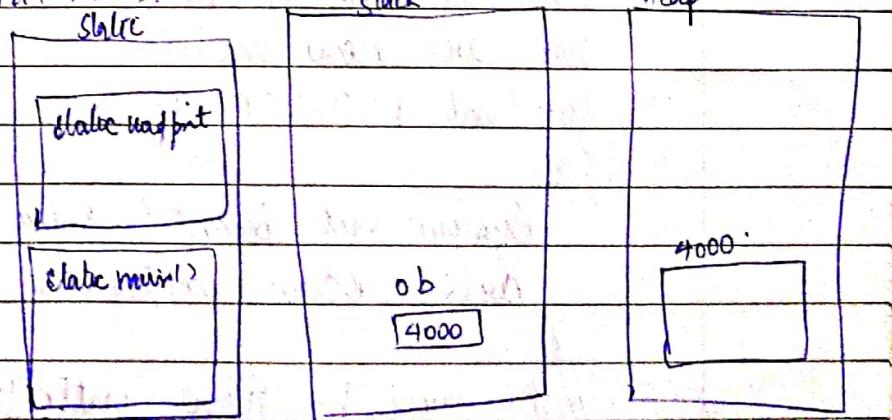
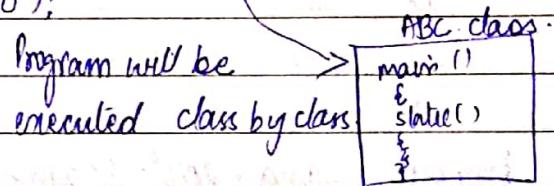
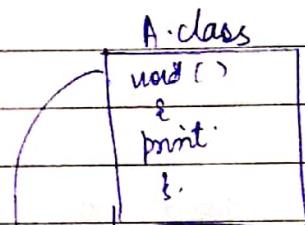
{

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

```
    A ob = new A();
    ob.print();
}
```

```
System.out.print ('ABC');
```

{



* Magic Number:

Good number: position of the numbers will not change after sorting

Bad number: position will change after sorting

Magic number: Good number - Bad number

SIP: arr [0 | 0 | 0 | 3 | 2 | 7]
 0 1 2 3 4 5

arr [0 | 0 | 0 | 2 | 3 | 7]
 0 1 2 3 4 5

$$\text{Good no: } 0+0+0+7 = 7$$

$$\text{Bad no: } 3+2=5$$

$$\text{Magic no: } 7-5=2$$

```
import java.util.Scanner;  
class MagicNumber  
{
```

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

```
        Scanner scan = new Scanner (System.in);  
        System.out.print ("Enter the size");
```

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

```
        int arr = new int[n];
```

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

```
    {
```

```
        System.out.print ("Enter the elements");
```

```
        arr[i] = scan.nextInt();
```

```
}
```

```
        int arr1[] = new int[n];
```

```
        Arrays.sort (arr);
```

while (read) {

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

```
{ arr[i] = arr[i]; }
```

```
}
```

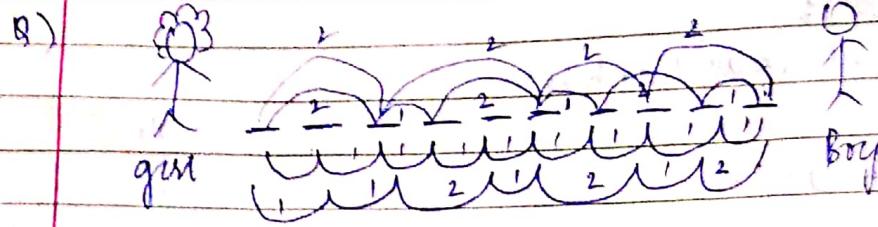
```
int goodnum = 0, badnum = 0;
```

b1:

arr [0|0|0|0|3|2|7]

arr [0|0|0|2|3|1|]

arr [0|0|0|6|3|7]



The boy can reach the girl in one step or two steps. Find the most efficient way for the boy to reach the girl.

- Q) factorial program: for larger inputs use long bigint data type-

- ⑧) public class Test {

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

3) System.out.print (2|0.0);

O/P: infinity (No exception error) Why?

Q ① Q/p: how are you all

O/p: How Are You All.

Q ② Q/p: → [1|2|7|6|2|3]

key - 8

O/p: 1 7

2 6

- ① Armstrong no.
② Prime no
③ factorial
④ reverse
⑤ JDB

$18 \times 2 - 2 \times 10^5$

1. String
2. prime no, factorial,
palindrome, fibonacci
series, number patterns

$36 - 2 \times 10^5$

1×10^5