

## Arithmetic operators

+

$$32154 + 32154 = 64308$$

-

$$32154 - 32154 = 0$$

\*

$$1 \times 10 = 10$$

/

$$20 / 2 = 10$$

%

Case-I

$$51 \% 2 = 1$$

$$100 \% 2 = 0$$

$$6 \% 2 = 0$$

$$5 \% 2 = 1$$

$$10 \% 4 = 2$$

Case-II

$$2 \% 5 = 2$$

$$22 \% .50 = 22$$

$$5 \% 3 = 5$$

Case-III

$$-5 \% 2 = -1$$

$$-5 \% 3 = -2$$

$$-5 \% 11 = -5$$

$$-n \% m = -(n \% m)$$

Case-IV

$$5 \% -2 = 1$$

$$5 \% -12 = 5$$

$$6 \% -4 = 2$$

Case-V

$$-100 \% 3 \Rightarrow -100 \% 3 = -1$$

$$-50 \% -101 \Rightarrow -50 \% 101 = -50$$

$$-n \% -m = -(n \% m)$$

$$n \% -m = n \% m$$

## Logical operators

&&

||

!

\*  $\text{! true} = \text{false}$

$\text{! false} = \text{true}$

$\text{!} 0 = 1$   
 $\text{!} 1 = 0$  X Bad operand types

&&

||

$\wedge$

A	B	y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

A	B	y
0	0	0
0	1	0
1	0	1
1	1	0

true && true && false  $\rightarrow$  false

false && true && false  $\rightarrow$  false

true && false && true && false  $\rightarrow$  false

\* S.O.P(1&&1) // Bad operand types

\* The operands should always be boolean

\* In 'Or' operator when it see first true condition, it will stop checking.

true || false || false  $\rightarrow$  true

false || false || true  $\rightarrow$  true

## Relational operators

<

>

$\leq$

$\geq$

$\neq$

$\equiv$

$2 \neq 5$  — True

$2 == 5$  — False

$5 < 5$  — False

$2 < 5$  — True

$5 > 6$  — False

$6 > 4$  — True

$5 \leq 5$  — True

$5 == 5$  — True

\* Always return Boolean values. [ $5 > 5 \rightarrow \text{False}$ ].

## Bitwise operators

\*  $6 = 110$

$13 = 1101$

$25 = 11001$

$$\begin{array}{r} 2 | 6 \\ 2 | 3 - 0 \\ \quad \quad \downarrow \\ \quad \quad 1 - 1 \end{array}$$

$$\begin{array}{r} 2 | 13 \\ 2 | 6 - 1 \\ 2 | 3 - 0 \\ \quad \quad \downarrow \\ \quad \quad 1 - 0 \end{array}$$

$$\begin{array}{r} 2 | 25 \\ 2 | 12 - 1 \\ 2 | 6 - 0 \\ 2 | 3 - 0 \\ \quad \quad \downarrow \\ \quad \quad 1 - 1 \end{array}$$

$$\begin{array}{r} 1100 \\ 0001 \\ \hline 1101 \end{array}$$

\* ~

?

^

!

$$\underline{348} \quad //0$$

$$\begin{array}{r} 0011 \\ + 1000 \\ \hline 0000 \end{array}$$

$$\underline{582} \quad //5$$

$$\begin{array}{r} 101 \\ 111 \\ \hline 101 \end{array}$$

$$\begin{array}{r} 215 \\ 213-1 \\ 1-0 \\ 217 \\ 213-1 \\ 1-1 \end{array}$$

$$\underline{13421} \quad //5$$

$$\begin{array}{r} 01101 \\ 10101 \\ \hline 00101 \end{array}$$

$$\begin{array}{r} 213 \\ 216-1 \\ 213-0 \\ \hline 1-1 \end{array}$$

$$\begin{array}{r} 2121 \\ 210-1 \\ 215-0 \\ 212-1 \\ \hline 1-0 \end{array}$$

\*  $\underline{517} \quad //7$

$$\begin{array}{r} 101 \\ 111 \\ \hline 111 \\ 2+2+2 = 7 \end{array}$$

$$\begin{array}{r} 215 \\ 213-1 \\ \hline 1-0 \end{array}$$

$$\begin{array}{r} 217 \\ 213-1 \\ \hline 1-1 \end{array}$$

\*  $\underline{13121} \quad //29$

$$\begin{array}{r} 01101 \\ 10101 \\ \hline 01101 \\ 16+8+9+1 = 29 \end{array}$$

\*  $\underline{318} \quad //11$

$$\begin{array}{r} 0011 \\ 1000 \\ \hline 1011 \end{array}$$

$$\begin{array}{r} 213 \\ 1-1 \end{array}$$

$$\begin{array}{r} 218 \\ 214-0 \\ 212-0 \\ \hline 1-0 \end{array}$$

Note:- If we use '^' bitwise operator, then their result should be less than or equal to the given operands.

\* In case of '^' bitwise operators, their result should be greater than or equal to the given operands.

$$* 5 \wedge 7 // 2$$

$$\begin{array}{r} 101 \\ \wedge 111 \\ \hline 010 \end{array}$$

$$* 7 \wedge 8 // 15$$

$$\begin{array}{r} 0111 \\ \wedge 1000 \\ \hline 1111 \end{array}$$

$$* 21 \wedge 3 // 22$$

$$\begin{array}{r} 10101 \\ \wedge 00011 \\ \hline 10110 \end{array}$$

$$16 + 0 + 4 + 2 + 0$$

$$\begin{array}{r} 2 | 10 - 1 \\ 2 | 5 - 0 \\ 2 | 2 - 1 \\ 1 - 0 \end{array}$$

Note: If we use 'f' bitwise operators, then their result should be less than or equal to the given operands.

\* In case of 'ox' bitwise operators, their result should be greater than or equal to the given operands.

\*  $5 \wedge 7 // 2$

$\begin{array}{r} 101 \\ \wedge 111 \\ \hline 010 \end{array}$

\*  $7 \wedge 8 // 15$

$\begin{array}{r} 0111 \\ \wedge 1000 \\ \hline 1111 \end{array}$

\*  $21 \wedge 3 // 22$

$\begin{array}{r} 10101 \\ \wedge 00011 \\ \hline 10110 \end{array}$

$\begin{array}{r} 210-1 \\ 212=1 \\ 1-0 \end{array}$

$16+4+2+0 = 22$

\* Method:

String B = Integer to Binary String (5); 5

S.O.P(5);

O|P:- 101

String B = Integer . to Binary string (0x5)

S.O.P(5)

O|P; 32-bits (4 bytes)

000-----00000101  
0 111-----11111010

if BLOCK:

if (1)

{  
S.O.P.C

?  
else

{ S.O  
?

\* In the

Syntax F

for C init

{

### Negations (~)

$$\begin{array}{r} 8 \ 4 \ 2 \ 1 \\ 1 \ 0 \ 0 \ 1 \end{array}$$

$$* \sim 99 = -(99+1)$$

$$\sim x = -(x+1)$$

$$\begin{array}{r} 0 \ 1 \ 1 \ 1 \\ 1 \ 0 \ 0 \ 0 \\ 1 \ 1 \ 1 \ 1 \Rightarrow 8 + 4 + 2 + 1 \end{array}$$

Eg:  $\sim 5$

$$\begin{array}{r} 0 \cdots 000101 \\ \sim 11 \cdots 111010 \\ \hline 0^3 + 2^1 \end{array}$$

$$= -8 + 2$$

$$= -6 //$$

$$O/P :- \underline{-6}$$

Eg:  $\sim 3 \rightarrow n3$

$$\begin{array}{r} 00 \cdots 00001001 \\ \sim 11 \cdots 11110110 \\ \hline 2^4 0 2^2 2^0 \end{array}$$

$$\begin{array}{r} -16 + 4 + 2 \\ \hline -10 // \end{array}$$

$$O/P :- \underline{-4}$$

Eg:  $\sim 9 \rightarrow n9$

$$\begin{array}{r} 00 \cdots 000001001 \\ \sim 11 \cdots 111110110 \\ \hline 2^4 0 2^2 2^0 \\ -16 + 4 + 2 \\ \hline -10 // \end{array}$$

$$O/P :- \underline{-10}$$

Eg:  $\sim 4 \rightarrow n13$

$$\begin{array}{r} 0000 \cdots 0001101 \\ \sim 1111 \cdots 1110010 \\ \hline 2^4 0 0 2^1 0 \\ -16 + 2 \\ = -14 \end{array}$$

Eg:  $\sim 5 \rightarrow -14 \ n13$

$$\begin{array}{r} 00 \cdots 00001101 \\ \sim 11 \cdots 1110010 \\ \hline 2^4 0 0 2^1 0 \\ -16 + 2 \\ = -14 \end{array}$$

### if BLOCK:

```
if (1)
{
    S.O.P ("Hai");
}
else
{
    S.O.P ("Bye");
}
```

- \* In the if condition we can only pass boolean values.

### SYNTHETIC FOR LOOP:-

for (Initialization; condition check; update)

  {  
    ~~initialization~~  
    ~~condition~~  
    ~~update~~  
    ~~body~~  
    ~~exit~~  
  };

true

false

Eg:-

for (int i=0; i<2; i++)

{  
    S.O.P (i);  
}

O/P :-  
=

Eg:-

for (int i=0; true; i++)

{  
    S.O.P (i);  
}

O/P :-

= 0.0.0

1

2

3

4

5

6

..

∞

Eg: for (int i=0; i<2; s.o.p(i))  
{  
    i++;  
}

O/p:-  
1  
2

Eg: for (int i=0; i<2; i++)  
{  
    s.o.p(i);

O/p:- Not defined anywhere.

Eg:

for (int i=0; i<2; s.o.p(i))  
i++;

s.o.p("incremented");

O/p:  
1  
2

incremented

for (int i=0; i<2; s.o.p(i))

= {  
    i++;  
}

s.o.p("incremented");

O/p:  
1  
2

incremented.

Eg: for ( int i=0; i<2; s.o.p(i))  
{  
    i++;

s.o.p("incremented");

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

## \* Class Launch

```
{ public static void main (String args []) {  
    { if (false);  
        {  
            System.out.println ("Hai");  
        }  
    }  
}
```

O/p :- Hai.

- If the condition false the if-block will be executed  
But since there is a semicolon after the condition  
that will end of if statement and block below it will  
treated as a normal block which as condition for it.
- Since it is not condition it will execute anyways.
- In the condition of for loop if you leave it empty,  
if you write it true <sup>(or)</sup> it is a constant ~~it is~~ this  
conditions will never changed .
- Throughout the condition always true the  
Compiler unreachable Statement for the  
Statement which are outside the for loop.

Ex:

## Class Launch

```
{  
    p s v m (String args[]){  
        for (int i=0; i<2; s.o.p(i))  
            i++;  
    }  
}
```

2

## Class Launch

```
{  
    p s v m (String args[]){  
        for (int i=0; i<3; i++)  
            for (int j=0; j<3; j++)  
                s.o.p(C[i][j]);  
    }  
}
```

**Output**

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

Final output of program is 0,1,2,3,4,5  
and we get object and class objects

		0	1	2	3	4
		(i,j)	$i+j = 0$	$i+j = 1$	$i+j = 2$	$i+j = 3$
i	0	0,0	0,1	0,2	0,3	0,4
	1	1,0	1,1	1,2	1,3	1,4
	2	2,0	2,1	2,2	2,3	2,4
	3	3,0	3,1	3,2	3,3	3,4
	4	4,0	4,1	4,2	4,3	4,4

### STAR PATTERN

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

{

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

{

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

{

```
            S.O.P(" * ");
```

}

```
        else
```

{

```
            S.O.P("   ");
```

}

```
S.O.P(C);
```

}

O/P :-

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



eg:  $i=0 \parallel j=0 \parallel i=n-1 \parallel i=n/2$

10 20 30 40 50

O/P:

\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*

M:

B C D E F H I L O T U

(+9 (-8 +7 (+6 -5) x)

(+9 (-8 +7 (+6 -5) x)

{ C1=0 || i=0 = 3 || 2=3 || 0=3 || 0=3 }

3 4 5 6 7 8

9 10

11 12 13 14 15

16 17 18 19 20

21

O/P:

□

\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*

eg:  $i=0 \parallel j=0 \parallel i=n-1 \parallel j=n/2$

Op:

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

HW

B C D E F H I L O T U

\* B

$i=0 \parallel j=0 \parallel i=n/2 \parallel j=n-1$

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

\* C

$i=0 \parallel j=0 \parallel i=n-1$

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

\* D

$i=0 \parallel j=0 \parallel i=n-1 \parallel j=n-1$

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

\* E

$i=0 \parallel j=0 \parallel i=n/2 \parallel j=n-1$

\* \* \* \* \*  
\*  
\* \* \* \* \*  
\*  
\* \* \* \* \*

\* F

$i=0 \parallel j=0 \parallel i=n/2$

\* \* \* \* \*  
\*  
\* \* \* \* \*  
\*  
\*

\* H

$j=0 \parallel i=n/2 \parallel j=n-1$

\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*

\* I

$i=0 \parallel i=n-1 \parallel j=n/2$

\* \* \* \* \*  
\*  
\*  
\*  
\* \* \* \* \*

\* L

$$i = n-1 \quad || \quad j = 0$$

\*  
\*  
\*  
\*  
\* \* \* \*

\* T

$$i = 0 \quad || \quad j = n/2$$

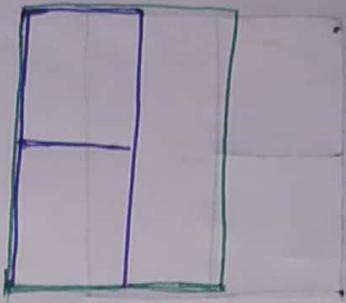
\* \* \* \* \*  
\*  
\*  
\*  
\*

\* U

$$i = n-1 \quad || \quad j = 0 \quad || \quad j = n-1$$

\* \* \* \* \*      \*  
\* \* \* \* \*      \*  
\* \* \* \* \*      \*  
\* \* \* \* \*      \*  
\* \* \* \* \*      \*

\* \* \* \* \*  
 \* \* \* \*  
 \* \* \* \*  
 \* \* \* \*  
 \* \* \* \*



$j = 0 \text{ || } j = n/2 \text{ || } i = 0 \text{ & } j < n/2 \text{ || } i = n/2 \text{ & } j < n/2$   
 $\text{|| } o = i$

Ans

For (int  $i = 0$ ;  $i < n$ ;  $i++$ ) { }

{

  For (int  $j = 0$ ;  $j < n$ ;  $j++$ ) {  $o = i$  ||  $o = -i$  }

{

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

{

    S.O.P ("\*");

}

  else

{

    S.O.P (" "); ( $o == i \text{ || } o == -i \text{ || } o == 0$ )

}

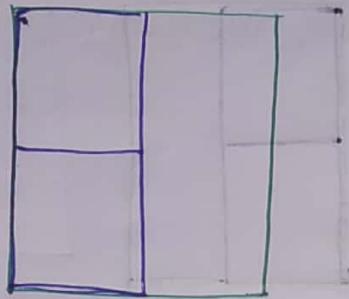
  System.out.println();

}

}

$i = i + 1 \text{ || } o = -i \text{ || } o = i$

\* B

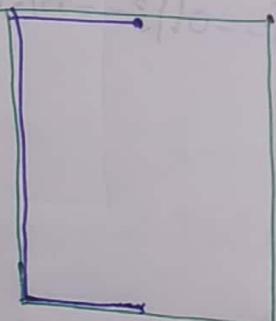


$j = 0 \quad || \quad j = n/2 \quad || \quad i = 0 \quad \& \quad j < n/2 \quad || \quad i = n/2 \quad \&$   
 $i = n-1 \quad \& \quad j = n/2 \quad || \quad j = n/2 \quad ||$

(or) 
$$\begin{aligned} & [a * b + c * b + d * b] \\ & [a + c + d] * b \end{aligned}$$

$j = 0 \quad || \quad j = n/2 \quad || \quad c \quad i = 0 \quad || \quad i = n/2 \quad || \quad i = n-1 \quad \& \quad j < n/2$

\* C

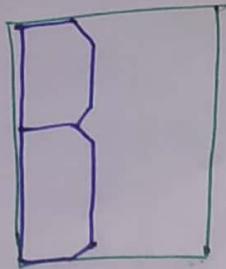


$j = 0 \quad || \quad (j = 0 \quad || \quad i = n-1), \quad \& \quad j < n/2$

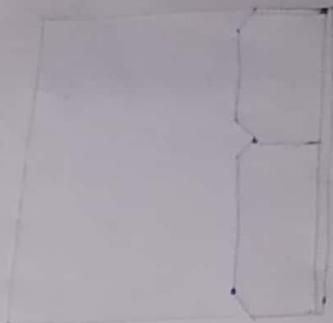
\* A

\*\*  
\* \*  
\*\*\* \*  
\* \*  
\* \*

$i = n/2 \quad || \quad j < n/2 \quad || \quad i = 0 \quad \& \quad j > 0 \quad || \quad i = n/2 \quad \& \quad j >= 1$



B



$\& i > 0$

$j = 0 \quad || \quad i = 0 \quad \& \& j < n/2 \quad || \quad j = n/2 \quad \& \& i < n/2 \quad ||$

$i = n-1 \quad \& \& j < n/2 \quad || \quad i = n/2 \quad \& \& j < n/2 \quad ||$

$j = n/2 + 1 \quad \& \& i > n/2 \quad \& \& i < n-1 \quad ||$

(08)

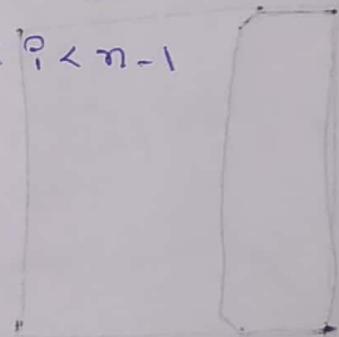
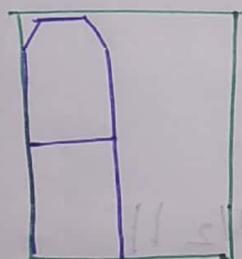
~~$i = 0 \quad || \quad j = 0 \quad \& \& i = 0 \quad || \quad i = n/2 \quad || \quad i = n-1 \quad \& \& j < n/2 \quad ||$~~

$j = n/2 \quad \& \& i > 0 \quad \& \& i < n/2$

$j = n/2 + 1 \quad \& \& i > n/2 \quad \& \& i < n-1$



A



$i = 0 \quad \& \& o = i \quad || \quad i = n-1 \quad \& \& o = i$

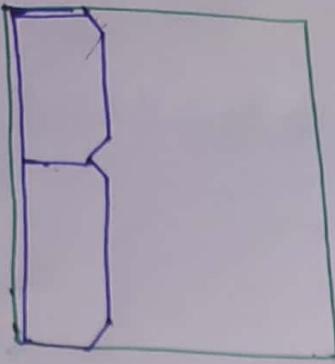
$j = 0 \quad \& \& i != 0 \quad ||$

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

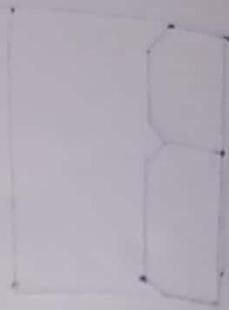
$i = n/2 \quad \& \& j < n/2 \quad ||$

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

\*



B



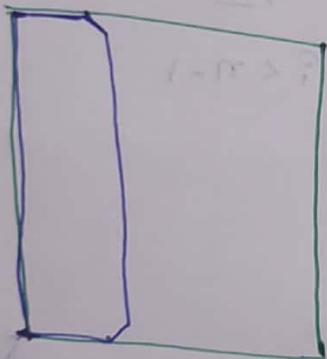
$$j = 0 \quad || \quad i = 0 + j < n/2 \quad \text{if } *$$

$$i = n/2 + j < n/2 \quad ||$$

$$i = n/2 + j < n/2 \quad ||$$

$$j = n/2 + i! = 0 + i! = n/2 + i! \approx n/2$$

\*



□

A

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

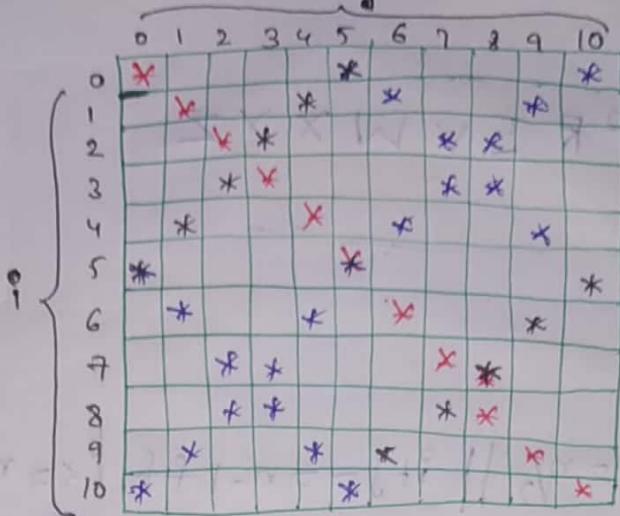
$$j = n/2 + i! = 0 \quad \text{if } i! = n-1$$

$$\text{if } i = 0 + i + 0 = 0$$

$$\text{if } i = 0 + i + 0 = 0$$

$$\text{if } i = 0 + i + 0 = 0$$

$$\text{if } i = 0 + i + 0 = 0$$



$$\begin{array}{|c|c|} \hline i & j \\ \hline 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ \hline i=j \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline i & j \\ \hline 0 & 10 \\ 1 & 9 \\ 2 & 8 \\ \hline \end{array}$$

$$i+j = n-1$$

$$\begin{array}{|c|c|} \hline i & j \\ \hline 5 & 0 \\ 4 & 1 \\ 3 & 2 \\ \hline \end{array}$$

$$i+j = n/2$$

$$\begin{array}{|c|c|} \hline i & j \\ \hline 0 & 5 \\ 1 & 6 \\ 2 & 7 \\ 3 & 8 \\ 4 & 9 \\ 5 & 10 \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline i & j \\ \hline 5 & 0 \\ 6 & 1 \\ 7 & 2 \\ \hline \end{array}$$

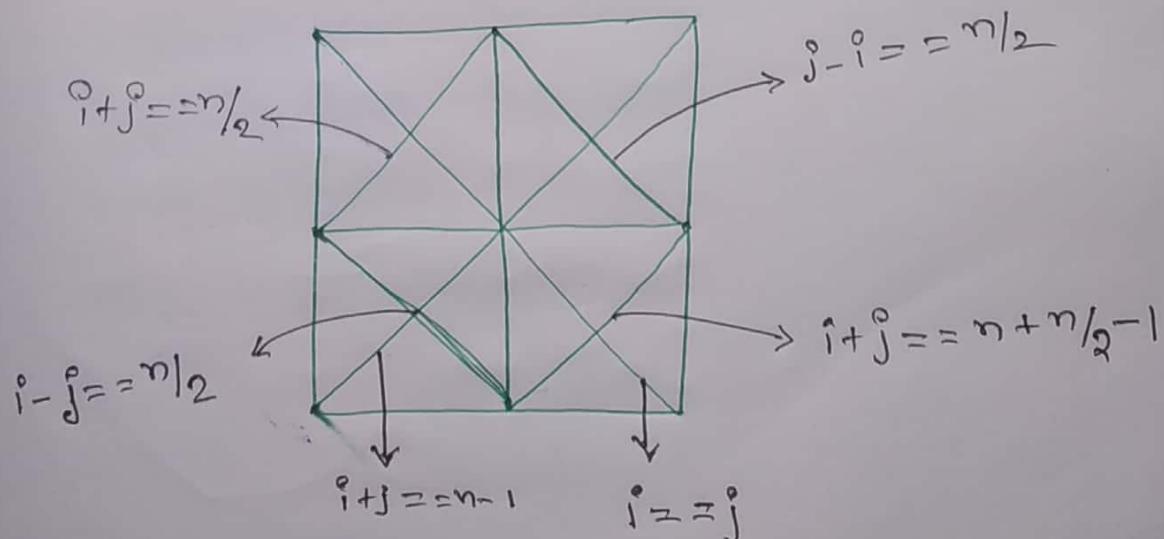
$$i-j = n/2$$

$$\begin{array}{|c|c|} \hline i & j \\ \hline 10 & 5 \\ 9 & 6 \\ 8 & 7 \\ \hline \end{array}$$

$$i+j = n+n/2-1$$

$$j-i = 5$$

$$j-i = n/2$$



HW

E G J K M N P R S V W X Y Z

$$i = j \text{ & } j \leq n/2 \quad || \quad i+j = n-1 \text{ & } i \leq n/2 \quad ||$$

$$j = n/2 \text{ & } i \geq n/2$$

(or)

$$(i = j \quad || \quad i+j = n-1) \text{ & } i \leq n/2$$

i	0
j	0
i	1
j	1
i	2
j	2
i	3
j	3
i	4
j	4
i	5
j	5
i	6
j	6
i	7
j	7
i	8
j	8

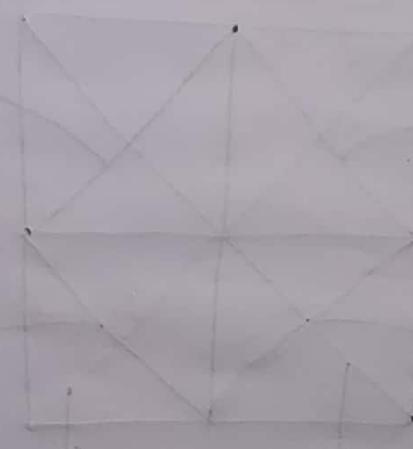
i	0
j	0
i	1
j	1
i	2
j	2
i	3
j	3
i	4
j	4
i	5
j	5
i	6
j	6
i	7
j	7
i	8
j	8

i	0
j	0
i	1
j	1
i	2
j	2
i	3
j	3
i	4
j	4
i	5
j	5
i	6
j	6
i	7
j	7
i	8
j	8

$$\sigma = i - 6$$

$$\sigma = 9 - 6$$

$$d\sigma = 9 - 6$$



HW

E G J K M N P R S V W X Y Z

\* Y

$i = j \text{ } \& \text{ } i \leq n/2 \quad \& \quad i+j = n-1 \text{ } \& \text{ } i \geq n/2$

$j = n/2 \text{ } \& \text{ } i \geq n/2$

(08)

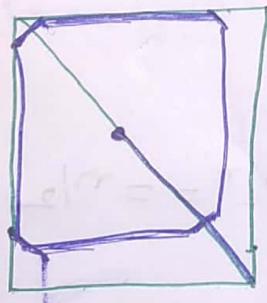
$(i = j \quad \& \quad i+j = n-1) \text{ } \& \text{ } i \leq n/2$

$j = n/2 \text{ } \& \text{ } i \geq n/2$

T

$i = 0 \quad \& \quad j = n/2 \quad \& \quad i-j = n/2 \quad \& \quad j >= n/4$

\*

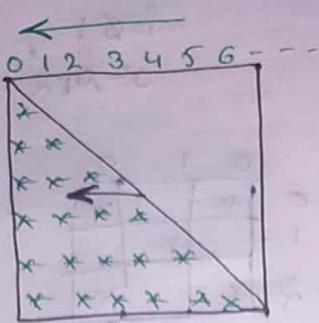


$i_1 = 0 \quad \& \quad i_1 = 3n/4 \quad \& \quad j_1 = 3n/4 \quad \& \quad j_1 = 3n/4$

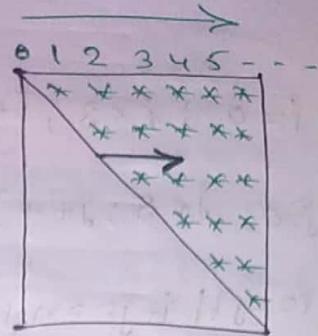
$j_1 = 0 \quad \& \quad i_1 = 0 \quad \& \quad i_1 = 3n/4 \quad \& \quad j_1 = 3n/4$

$i_1 = n-1 \quad \& \quad j_1 = 0 \quad \& \quad j_1 = n-1 \quad \&$

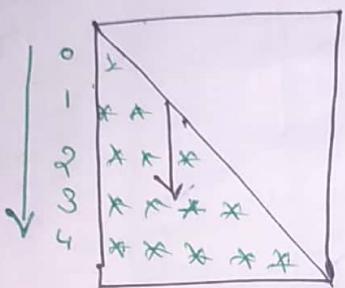
$j_1 = 3n/4 \quad \& \quad i_1 = 0 \quad \& \quad i_1 = 3n/4$



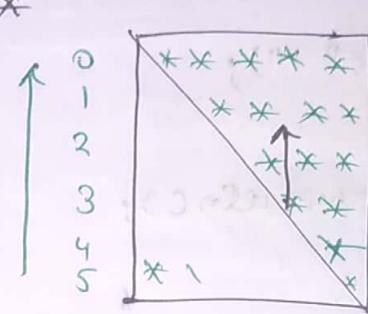
$$i \leq j$$



$$i \geq j$$



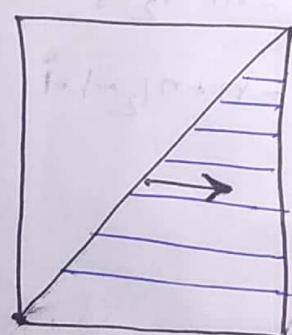
$$i \geq j$$



$$i \leq j$$

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

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



$$i+j \geq n-1$$

$$j \leq i$$

$$\begin{aligned} i+j &\geq n-1 \\ i &\geq n-1-j \\ i &\geq 1-j \end{aligned}$$

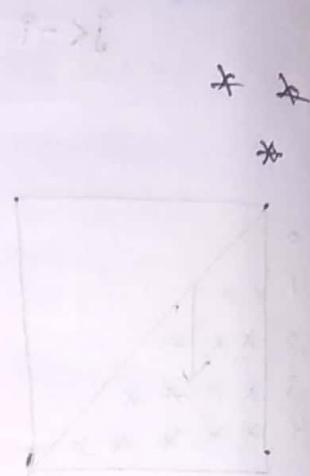
\* nos

```

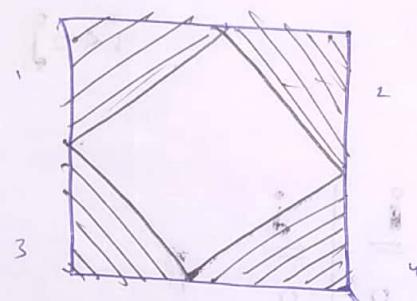
for<int i=0 ; i<3 ; i++)
{
    for<int j=0 ; j<3 ; j++)
    {
        if(i==j || i+j==n-1)
        {
            s.out.println("X");
        }
        else
        {
            s.out.println(" ");
        }
    }
    s.out.println();
}

```

	$i = \emptyset 1$
	$j = \emptyset 2$
0	*
1	
2	*



\*



$$\textcircled{1} \quad i+j = n_2 \quad \textcircled{2} \quad j-i = n_2$$

$$j <= n_2 - i \quad \textcircled{1} \quad j >= n_2 + i \quad \textcircled{2}$$

$$\textcircled{3} \quad i-j = n_2$$

$$i = j <= n_2$$

$$i = n_2 + j$$

$$i - n_2 = j$$

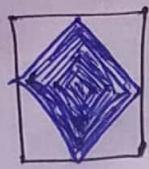
$$j <= i - n_2 \quad \textcircled{1}$$

$$\textcircled{4} \quad i+j = n+n_2 - 1$$

$$j >= n+n_2 - 1 - i$$

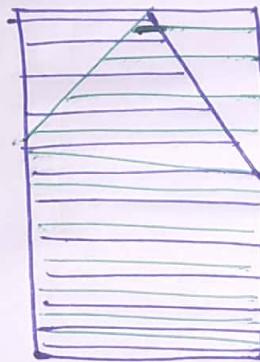


$$i - 1 - n_2 < b$$

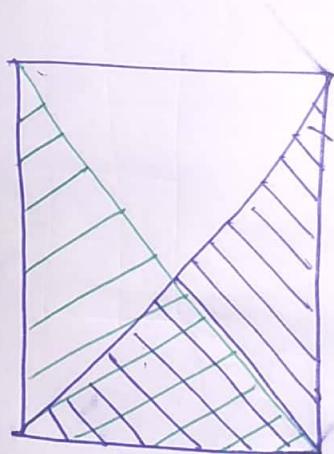


$$j>=n/2 - 1 \text{ if } j<=n/2 + 1 \text{ and } j<=n+n/2 - 1 \text{ if } j>=n-n/2$$

$$j>=n/2 - 1 \text{ if } j<=n/2 + 1$$



$$\text{Op}_i = \begin{cases} \text{square} & i=0 \\ \text{triangle} & i=1 \end{cases}$$



$$j>=n-1-i \text{ if } j=0$$

$$\text{Op}_i = \begin{cases} \text{square} & i=0 \\ \text{triangle} & i=1 \end{cases}$$

\*

\*\*\*

\*\*x

x\*\*

\$ \$\$

\$ \$ \$

Class Box      \$ \$ \$

{

s.s.m (Carrying argC)

for (i=0; i<3; i++)

{

shape(i++);

for(j=0; j<3; j++)

{

s.o.pc("x");

{

}

for (i=0; i<3; i++)

{

for(j=0; j<3; j++)

{

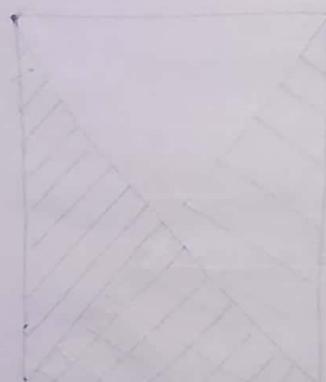
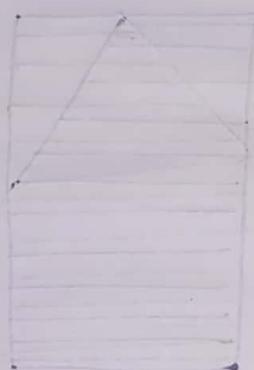
s.o.pc("\$");

{

s.o.println();

{

{}



\*

\* \* \* \$ \$\$

\* \* \* \$ \$ \$

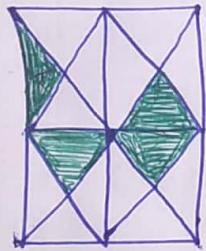
\* \* \* \$ \$ \$

```

int n = 3;
for (int i=0; i<3; i++)
{
    for (int j=0; j<3; j++)
        System.out.print ("*");
    System.out.println ();
}
for (int j=0; j<3; j++)
{
    System.out.print ("$");
}
System.out.println ();

```

HW



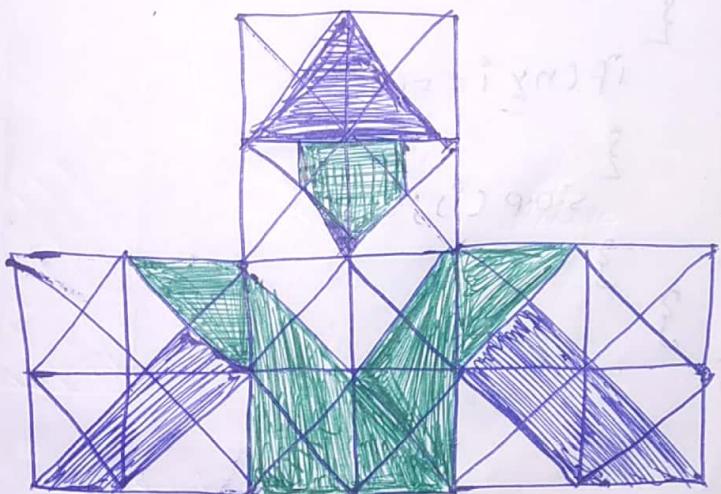
①

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

$$j < i$$

$$j \leq n - 1 - i$$

=



②

6	G	6	G	G	6	→ X	6	6	6	6	→ n
X	X	X	X	X	X		X	X	X	X	
1	2	3	4	5	6	7	8	9	10.	→ r	
11	11	11	11	11	11		11	11	11	11	
0	0	0	2	1	0	6	6	6	6	→ R	
✓	✓	✓	X	X	✓	X	X	X	X		

## \* Prime. no. Program 1.1

Class Prime

{

PSVM (String args[])

{

int n=6;

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

{

if (n% i==0)

{

S.O.P (i);

}

}

O/P :- 2, 3, 6

\* Program-2 :- count the number of prime factors.

```
int n=5;  
int c=0;
```

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

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

```
{  
    c++  
}
```

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

```
}
```

\* Program-3

```
int n=5;
```

```
int c=0;
```

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

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

```
{  
    c++;  
}
```

```
}  
if (c == 2)
```

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

```
}  
else
```

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

### Program-1

Class Haxi

{

Psum(String args[])

{

int n = 5;

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

{

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

{

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

}

S.0. Pm();

} }

O/P

0 0 0 0 0

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

### Program-2

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

i(have)	want(w)
0	4
1	3
2	2
3	2
4	0

$$i + w = 4$$

$$i + w = 5 - 1$$

$$i + w = 5 - 1$$

$$w = 5 - 1 - i$$

Class Launch

```
{  
    psum(String args[]){  
        int n=5;  
        for (int i=0; i<n; i++) {  
            for (int j=0; j<n; j++) {  
                System.out.println(i+j);  
            }  
        }  
    }  
}
```

0|P:  
=

0|P:  
=

Program - 8

Class Launch

```
{  
    psum (String args[]){  
        int n=5;  
        for (i=0; i<n; i++) {  
            for (j=0; j<n; j++) {  
                System.out.println(j%2);  
            }  
        }  
    }  
}
```

0|P:  
=

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

0|P:  
=

## Programs - 4

```

Class Launch
{
    public static void main(String args[])
    {
        int n = 5;
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                System.out.println((i + j) % 2);
            }
        }
    }
}

```

$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ \hline 3 & 0 & 1 & 0 \\ 4 & 1 & 0 & 1 \\ \hline \end{array}$

$$\underline{(j+1) \% 2}$$

## Program - 5

```

Class Hari
{
    public static void main(String args[])
    {
        int n = 5;
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                System.out.println((i + j) % 2);
            }
        }
    }
}

```

$\begin{array}{c} j \\ \hline \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \hline \begin{array}{ccccc} 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 \\ 3 & 1 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 1 \end{array} \end{array}$

$$\underline{(i+j) \% 2}$$

### Program-6

$$(i+j+1) \% 2$$

$i \leftarrow$ 

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

 $\xrightarrow{\text{OP}}$

### Program-7

class Haxi

{ public static void main(String args[])

```

    {
        int n=5;
        int c=1;
        for(int i=0; i<n; i++)
    }

```

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

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

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

```
}
```

### Program-8

```
int n=5;
```

```
int c=1;
```

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

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

```
    if (j==i)
```

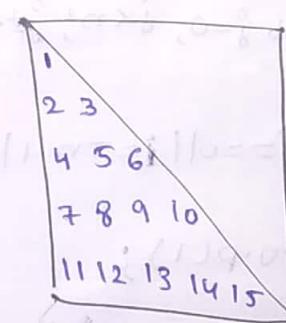
```
{
```

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

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

```
}
```

$\xrightarrow{\text{OP}}$  cursor movement  
 1 2 3 4 5  
 6 7 8 9 10  
 11 12 13 14 15  
 16 17 18 19 20  
 21 22 23 24 25



### Program - 9

```

int i=5;
int c=1;
for (int i=0; i<n; i++)
{
    for (int j=0; j<n; j++)
    {
        if (j<=i)
            cout << i % 2;
    }
}

```

O/P

0  
0 1  
0 1 0  
0 1 0 1  
0 1 0 1 0  
0 1 2 3 4 → j

### Program - 10

```

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

```

O/P

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

```

if (i==0 || i==n-1 || j==0 || j==n-1)
{
    cout << 1;
}
else
{
    cout << 0;
}

```

## program-11

```
int n=5;  
for (int i=0; i<n; i++)  
{  
    for (int j=0; j<n; j++)  
    {  
        if (j<=i)  
        {  
            s.o.p(j);  
        }  
        else  
        {  
            s.o.p(" ");  
        }  
    }  
  
    for (int j=1; j<n; j++)  
    {  
        if (j>=n-1-i)  
        {  
            s.o.p(n-1-j);  
        }  
        else  
        {  
            s.o.p(" ");  
        }  
    }  
  
    s.o.pn();  
}
```

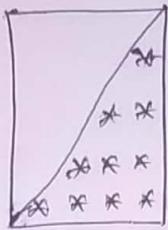
0	0
0 1	1 0
0 1 2	2 1 0
0 1 2 3	3 2 1 0
0 1 2 3 4	3 2 1 0

## Program 12

mandatory  
spaces

```

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



$$j \geq n - i - 1$$

remove the

spaces

```

int n=5;
for( int i=0; i<n; i++)
{

```

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

```

```
    if (j >= n - i - 1)
  {

```

```
    s.o.p(" * ");
  }
  else
  {

```

```
    s.o.p("  ");
  }
}
s.o.pln();
}

```

O/P :-

```

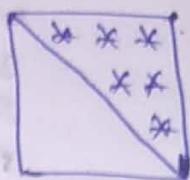
      *
      * - *
      * - * - *
      * - * - * - *
      * - * - * - *
    
```

### Program-13

```

int n=3;
for (int i=0; i<n; i++)
{
    for (int j=0; j<n; j++)
    {
        if (j>=i)
        {
            s.o.p(" * ");
        }
        else
        {
            s.o.p("  ");
        }
    }
    s.o.println();
}

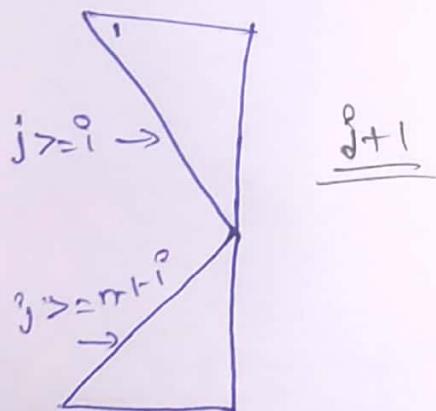
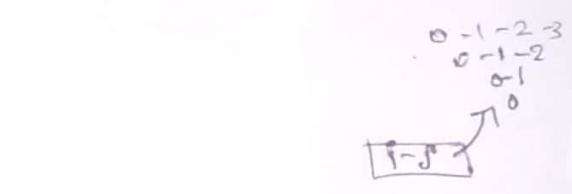
```



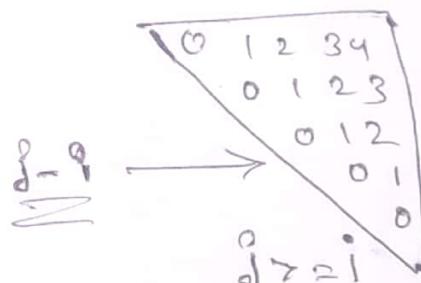
$$j >= i$$

$\times - \ast - \ast$   
 $- \ast - \ast$   
 $- - \ast$

### Program-14



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

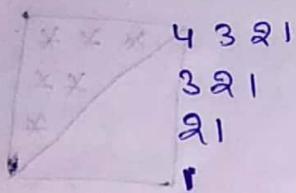


$$j = = i$$

$$i - j$$

## \* Programs

5 4 3 2 1



$$j = n - 1 - i$$

$$j - (n - 1 - i) \quad | \quad (n - 1 - i) - j$$

HQ

2  
2 3  
2 3 4  
2 3 4 5  
2 3 4 5 6

s.o.p( " " );

				2
		2	3	
	2	3	4	
2	3	4	5	
2	3	4	5	6

$$j >= n - 1 - i$$

\* 1 \* 2 \* 3 \* 4

5 \* 6 \* 7 \* 8

9 \* 10 \* 11 \* 12

13 \* 14 \* 15 \* 16

\*

```
int n=4;
int c=1;
for( int i=0; i<n; i++ )
{
```

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

```
{
```

```
    if( j==n-1 )
```

```
{
```

```
        s.o.p( c++ );
```

```
}
```

```
else
```

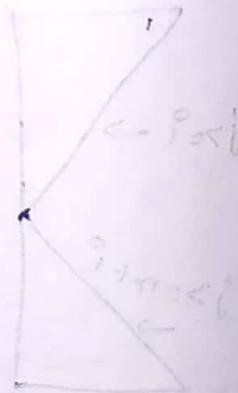
```
{
```

```
    s.o.p( c++ + " " );
```

```
}
```

```
} }
```

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



\*

1

2 \* 3

4 \* 5 \* 6

7 \* 8 \* 9 \* 10

int h = 4

int c = 1

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

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

{ if (j <= i)

{

group(c) →  
else  
if

if (j == i)

{

s.o.p(c++);

{

else

{

s.o.p(c++ + "\*");

} } } } } s.o.pmc();

\*

n=4.

\*

0 1

\*

2 3

\*

4 5

\*

6 7

\*

8 9

\*

10 11

\*

12 13

\*

14 15

\*

16 17

c = i+1

c = j+1

c = i+1

c = j+1

" "

(i+j+1) %

(j+1) %

(i+1) %

(j+1) %

" "

m=4.

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

{ int c = i+1;

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

{

s.o.p(c++ " ");

c = c+n

3

s.o.pmc();

}

m=4

i ≠ 1

j ≠ 1

c ≠ 1

	+5x0	+5x1	+5x2	+5x3	+5x4
i=0	1	6	11	16	21
i=1	2	7	12	17	22
i=2	3	8	13	18	23
i=3	4	9	14	19	24
i=4	5	10	15	20	25

52n

0, 1, 2, 3, 4 =  $\frac{0}{5}$

Put n=5

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

{

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

{

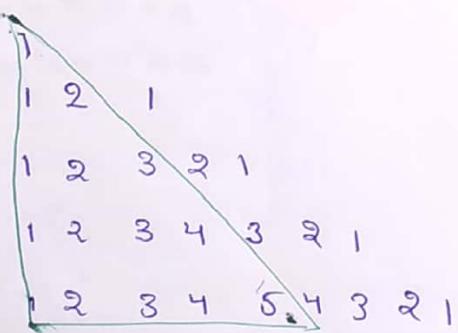
```
        s.o.p((i+1)+(n*j));
```

}

```
    s.o.pn();
```

}

\*



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

{

```
    s.o.p((j+1));
```

}

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

{

```
    s.o.p((i-j+1));
```

}

0  
1 0  
2 1 0  
3 2 1 0

4 3 2 1 0

	0	1	2	3	4	5	6
→	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0
2	0	1	2	2	2	1	0
3	0	1	2	3	2	1	0
4	0	1	2	2	2	1	0
5	0	1	1	1	1	1	0
6	0	0	0	0	0	0	0

$$\begin{array}{|c|c}
 \hline j & w \\
 \hline 6 & 0 \\
 5 & 1 \\
 4 & 2 \\
 \hline
 \end{array}$$

$j + w = 6$   
 $j + w = n - 1$   
 $w = -n - 1 - j$

$\text{int } n = 7;$

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

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

{  
    if ( $j >= i \& j <= n - 1 - i$ )

    s.o.p(i);

}

  else if ( $j >= i + 4 \& j <= n - 1 - i$ )

    s.o.p(n - 1 - i);

}

  else if ( $j <= i + 4 \& j <= n - 1 - i$ )

    s.o.p(j);

}

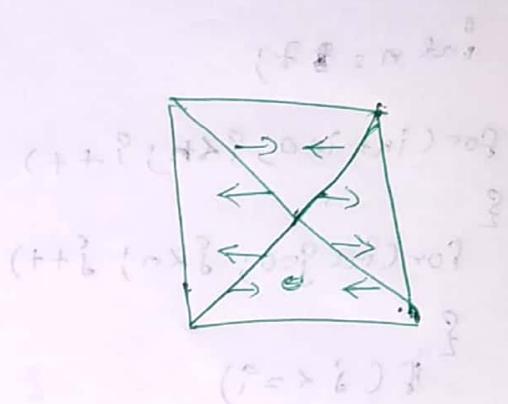
  else

    s.o.p(n - 1 - i);

}

  s.o.p(m);

}



\*   
 1  
 1 1  
 1 0 1  
 1 0 0 1  
 1 0 0 0 1  
 1 0 0 0 0 1  
 1 1 1 1 1 1 1  
 0 0 0 0 0 0 0

`int n = 8; }`

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

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

`{ if (j <= i)`

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

`{ s.o.p ("1");`

`}`

`else`

`{ s.o.p ("0");`

`}`

`}`

`} s.o.pln();`

\* 0 1

1 3 2

2 6 5 4

3 10 9 8 7

0 1 2 3

$m = 1$

$m = 0 + 1$

$m = 2$

$m = 3$

$m = 4$

int n = 4;

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

{ int m = i+1; int sn =  $\frac{m \times (m+1)}{2}$ ;  
for(int j=0; j<n; j++)

{

if(j <= i)

{

cout << sn+j + " "

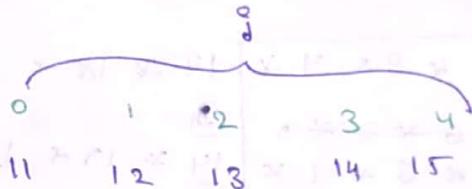
} {

System.out.println();

}

$$S_n = \frac{m(m+1)}{2}$$

i=0 m=4 10+1



i=1 m=3 6+1



i=2 m=2 3+1



i=3 m=1 1+1



i=4 m=0 0+1



int n=5

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

{

int m = n-i-1;

int sn =  $\frac{(m \times (m+1))}{2} + 1$ ;

{ if(j <= n-1-i)

{

cout << sn+j + " "

} {

cout << ln();

}

$$i+m = m-1$$

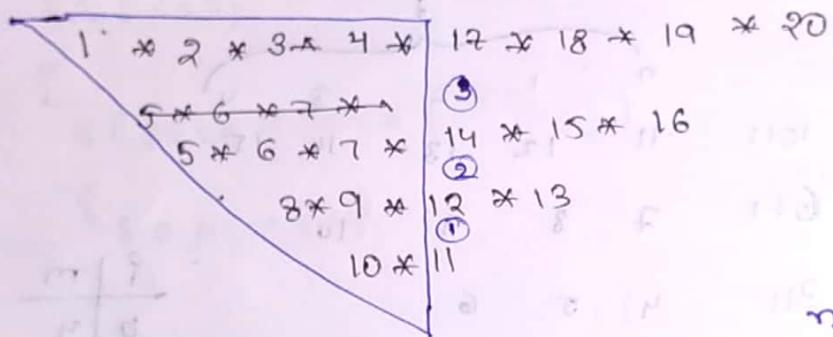
$$m-1-j = m-1$$

$$(i+m) \geq m-1$$

$$(i+m) \geq m-1$$

$$(i+m) \geq m-1$$

\*   
 0  
 0 1 0  
 0 1 2 1 0  
 0 1 2 3 2 1 0  
 - 0 1 2 1 0  
 6 1 0  
 0



$\text{int } n = 4;$

$\text{int } d = n - 1;$

$\text{int } s_n = n * n + 1;$

$\text{int } c = 1;$

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

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

{  $\text{if } (j \geq i)$

{

```

S.y stem.out.print( ctt + " * ");
}
else
{
    s.o.p("  ");
}

if ( j <= n - i - 1 )
{
    if ( j == n - i - 1 )
    {
        s.o.p(snt+j);
    }
    else
    {
        s.o.p(snt+j + " * ");
    }
}

s.o.println();
del s = s - d;
d--;
}
}

```

	0	1	2	3	4
0	1	1	1	1	2
1	3	2	2	2	2
2	3	3	3	3	4
3	5	4	4	4	4
4	5	5	5	5	6

even row =  $i+2$   
odd row =  $i+1$

```
int n=5;
```

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

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

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

```
{  
    if(j == n-1)
```

```
{  
    s.o.p(i+2);
```

```
}
```

```
else
```

```
{  
    s.o.p(i+1);
```

```
}
```

```
else
```

```
{  
    s.o.p()
```

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

```
{  
    s.o.p(i+2);
```

```
}
```

```
else
```

```
{  
    s.o.p(i+1);
```

```
} } }
```

```
s.o.p();
```

```
}
```

0	1	2	3	4			
*	0	1	2	3	4	5	$(5 \times 0) + 1$
	1	10	9	8	7	6	$(5 \times 2)$
2	11	12	13	14	15		$(5 \times 2) + 1$
3	20	19	18	17	16		$[5 \times 4]$
4	21	22	23	24	25		$(5 \times 4) + 1$

int n=5;

```

for (int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        if (i+j == 0)
        {
            s.o.printf ("%d", n*i + 1 + j);
        }
        else
        {
            s.o.printf ("%d", n*(i+1)-j);
        }
    }
    s.o.println();
}

```

④

0	1					
2	3	4	5	6		
10	9	8	7			
11	12	13	14	15		
21	20	19	18	17	16	

$\sum x_0 + 1 -$

int n = 5; (char) 1+(0x30) 8 11 8 1 9 0

(0x73) 2 F 8 F 0 1

for( int i=0; i<n; i++ ) 1+(0x30) 71 44 81 61 11 5

{ (char) 21 81 81 81 81 81 8

  for( int j=0; j<n; j++ ) 1+(0x30) 73 80 80 80 16 4

{

⑯	i = 0	m = 1	1
	i = 1	m = 2	3 2
	i = 2	m = 3	6 5 4
	i = 3	m = 4	10 9 8 7
	i = 4	m = 5	15 14 13 12 11

j = 0, 1, 2, 3, 4

$$m \times (m+1) = \frac{4 \times (4+1)}{2} = \frac{20}{2}$$

$$m \times 3 = \frac{3 \times 4 \times 2}{2} = 6$$

int n = 5;

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

{

  int m = i+1;

  int sn = m \* (m+1)/2;

  for( int j=0; j<n; j++ )

{

    if (j <= i)

      sn = sn - j + 1;

(2)      0 2 0 2  
 1 8 9 16  
 2 7 10 15  
 3 6 11 14  
 4 5 12 13

odd column }  $(m \times j) + 1 + i$

even column }  $n \times (j+1) - i$

\* System.out.println('a'+0);

O/p :- 97

A - 65	a - 97
Z - 90	z - 122

\* System.out.println((char) 65);

O/p :- A

\* System.out.println((char) ('A'+1));

O/p :- B

\* System.out.println((char) ('B'+32));

O/p :- ~~b~~ b

\* System.out.println((char) ('a'-32));

O/p :- A

\* char c = '1'      char to int  
 System.out.println(c);

O/p :- 1 integer

\* Int n=4

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

{

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

{

s.o.p ((char)(i+65));

}

s.o.pln();

}

O/P :-

A A A A

B B B B

C C C C

D D D D

→ s.o.p ((char) 65); // A

→ s.o.p ((int) 'A'); // 65 → ASCII values

→ s.o.p ('A' + 1); // 66 → char → int converted

→ s.o.p ((char) ('A'+1)); // B

→ char c = '1';

if (c >= 48 && c <= 57)

{

s.o.pln ("number");

}

else

{

s.o.pln ("Not a Number");

}

O/P :- Numbers

→ char c = '1'

if (c >= 65 && c <= 90 || c >= 97 && c <= 122)

{ s.o.p ("Alphabet");

}

else s.o.p ("Not a Alphabet"); }

```

→ charc = '@';
if (c >= 65 & & c <= 90) // c >= 48 & & c <= 57 || c >= 97 & & c <= 122)
{
    s.o.pnC("not a special character");
}
else
{
    s.o.pnC("special character");
}
}

```

O/P :- special characters

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

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

```

* charc = ' ';
if (c >= 65 & & c <= 90) // c = 'A' & & c = 'Z'
{
    s.o.p("UC");
}
else if (c >= 97 & & c <= 122) // c = 'a' & & c = 'z' space = 32
{
    s.o.p("LC");
}
else if (c >= 48 & & c <= 57) // c = '0' & & c = '9'
{
    s.o.p("Number");
}
else
{
    s.o.p("special character");
}

```

O/P :- special characters

- \* s.o.p (Character. is Digit ('2'));
- \* s.o.p (Character. is Digit ('0'));
- \* s.o.p (Character. is Digit ('E'));
- \* s.o.p (Character. is Digit ('\*'));
- \* s.o.p (Character. is UpperCase ('A'));
- \* s.o.p (Character. is UpperCase ('g'));
- \* s.o.p (Character. is UpperCase ('A'));
- \* s.o.p (Character. is LowerCase ('s'));
- \* s.o.p (Character. is LowerCase ('a'));
- \* s.o.p (Character. is LowerCase ('A'));
- \* s.o.p (Character. to LowerCase ('a'));
- \* s.o.p (Character. to LowerCase ('A'));
- \* s.o.p (Character. to LowerCase ('s'));
- \* s.o.p (Character. to UpperCase ('g'));
- \* s.o.p (Character. is Letter ('s'));
- \* s.o.p (Character. is Letter ('A'));
- \* s.o.p (Character. is Letter ('\*'));
- \* s.o.p (Character. is Letter Or Digit ('1'));
- \* s.o.p (Character. is Letter Or Digit ('a'));
- \* s.o.p (Character. is Letter Or Digit ('e'));

## Programs

```
import java.util.Scanner;  
class Demo  
{  
    public static void main (String args [])  
    {  
        Scanner scan = new Scanner (System.in)  
        String f = scan.nextLine(); // user enter first String  
        String s = scan.nextLine(); // user enter second String  
        if (f.contains(s))  
        {  
            System.out.println ("Found");  
        }  
        else  
        {  
            System.out.println ("Not Found");  
        }  
    }  
}
```

$a^*$  → \* 0 or more times

$a^+$  → + 1 or more times

$a?$  → ? 0 or 1 time

$a\{3\}$  → {3} exactly 3 times

$a\{3,6\}$  → {3,6} exactly from 3 to 5

[abc] → [ ] matching any character

[0-9]

[a-z]

[A-Z]

[^a-zA-Z0-9]

### \* Replace

String s = "abaca";

String t = s.replace("a", "z");

s.o.p(t);

O/P :- zbzcz

### \* Delete all character from a string

String s = "abcaacaa";

String t = s.replace("a", "");

s.o.p(t);

→ Empty String

O/P :- bcc

\* write a program to delete all lowercase alphabets from a given string.

String s = "AaBbCcDdFf\$";

String t = s.replaceAll("[a-z]", "");

s.o.p(t);

O/P :- ABHf3\$

\* write a program to copy lowercase alphabets.

String s = "AaBbCcDdHh\$";

String t = s.replaceAll("[a-z]", "");

s.o.p(t);

O/P :- accd

\* To copy Special characters

String s = "aA bB & 2 c#";

String t = s.replaceAll("[0-9 a-zA-Z]", "");  
s.o.p(t);

O/P :- & #

→ • → Any Characters [Dot(.) means any character]

String s = "abc12bc34";

String t = s.replaceAll(".[0-9]{2}", "");  
s.o.p(t);

O/P :- abb

→ String s = "abcd";  
o1234

s.o.p(s.charAt(0));  
s.o.p(s.charAt(1));  
s.o.p(s.charAt(2));  
s.o.p(s.charAt(3));  
s.o.p(s.charAt(4));

O/P :- abcd a

→ String s = "abcd"

for (int i=0; i<3; i++)  
{  
s.o.p ln(s.charAt(i));  
}

O/P :-  
a  
b  
c

→ Reverse      String

String,  $s = "haxi"$ ;

for (int  $i=3$ ;  $i \geq 0$ ;  $i--$ )

{

  s.o.println(s.charAt(i));

}

O/P :-

i

x

a

h

→ length()

a → s.length() - 1 = Last Index

o

a

b

i

↓

o

a

b

c

i

2

a

b

c

d

o

1

2

3

$s = "g b c d e f g h i j"$

s.o.p(s.length() - 1);

O/P =  $j$

⇒ string  $s = "abc"$ ;

for (int  $i = s.length() - 1$ ;  $i \geq 0$ ;  $i--$ )

{

  s.o.p(s.charAt(i));

}

  s.o.println();

}

O/P :-  
= cba

## \* Reverse a String

{ String s = "Haskell";

String t = "";

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

{

~~s[i]~~ t = t + s.charAt(i);

}

s.out(t);

}

O/P :- i8aH

## \* t = s.charAt(i) + t

t = 'a' + "

t = "a"

"~~a~~" = 'b' + "

b = "ba"

"~~ba~~" = 'c' + " "

b = "abca"

0 ≤ 3
1 ≤ 3
2 ≤ 3
3 ≤ 3

t = "~~cba~~"

t = "dcba"

O/P :- abedc - dcba

\* String s = "abcd";

String t = " ";

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

{

~~t~~ = ~~s.charAt(i)~~ + t;

}

s.outn(t);

}

## Palindrome

String s = "abc";

String t = "t";

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

{

    t = s.charAt(i) + t;

}

    if (t.equals(cs))

{

        s.o.p("Palindrome");

}

else

{

        s.o.p("Not a Palindrome");

}

O/P :- Not a Palindrome

\* 18th class

a b b a  
0 1 2 3

si = 0 x 3  
ei = 3 x 1

a b c d c b a  
0 1 2 3 4 5 6

si = 0 x 3  
ei = 6 x 3

int a=10;

int b=20;

}

int a=10, b=20;

## Program

\* Given  $s = "abcba"$ ;

```

int c = 0;
for (int si=0, ei=s.length()-1; si<ei; si++, ei--) {
    if (s.charAt(si) == s.charAt(ei))
        c++;
    else
        break;
}
if (c == 0)
    s.o.p("palindrome");
else
    s.o.p("! a palindrome");
}
}

```

O/P :- Palindrome.

\* Write a program to reverse String:

```

String s = "abcd";
StringBuffer sb = new StringBuffer(s);
sb.reverse();
s.o.p(sb.toString());
}
}

```

O/P :- dcba

```

String s = "abba";
StringBuffer sb = new StringBuffer(s);
if (s.equals(sb.reverse()))
{
    System.out.println("palindrome");
}
else
{
    System.out.println("not a palindrome");
}

```

dp8 palindrome

String + int addition :-

$$\Rightarrow 3 + 4 + "hai" + 4 + 5$$

$$7 + "hai" + 4 + 5$$

$$"7hai" + 4 + 5$$

$$"7hai4" + 5$$

$$"7hai45"$$

$\Rightarrow 1 \ 2 \ 3 \ CD!@$

~~use replaceAll("[0-9]@")~~

123 abcD!@

String t = s.replaceAll("[^0-9]", "");

String p = s.replaceAll("[^0-9]", "");

String ar = (t + " " + p);

\* String s = 1a2b3cD!@

123 abCD!@ → o/p

String num = s.replaceAll("[0-9]", "4");

String nonnum = s.replaceAll("[0-9]", "");

String f = (num + " " + nonnum);

s.o.p(f);

\* without built-in method

String num = "v";

String nonnum = "u";

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

{ if (s.charAt(i) >= 48 && s.charAt(i) <= 57)

{ num = num + s.charAt(i);

} else

{ nonnum = nonnum + s.charAt(i);

}

s.o.p (num + " " + nonnum);

o/p

123 abCD!@

\* String s = "1Har!"

StringBuffer sb = new StringBuffer(s);

s.o.p(sb.deleteCharAt(0));

o/p → Har!

Input :- Apple

Output :- Aple

\* Program to remove duplicates in a string :-

import java.util.Scanner;  
class Lec10

{

    public static void main(String args[]){

{

        Scanner scan = new Scanner(System.in);  
        String s = scan.nextLine();

        String t = "";

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

{

            if(t.contains(s.charAt(i)))

{

}

        else

{

            t = t + s.charAt(i);

}

}

        System.out.println(t);

}

O/P :-

APLe

\* Program to remove duplicates

class Launch

{  
    PSVM (String args[])

{

    static boolean PresentOrNot (char c, String s)

{

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

{

            if (c == s.charAt(i))

{

                return true;

} {

            return false;

}

PSVM (String args[])

{

    Scanner scan = new Scanner (System.in);

    String s = scan.nextLine();

    String t = " ";

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

{

        if (!PresentOrNot (s.charAt(i), t))

{

        else

{

            t = t + s.charAt(i);

}

}

    s.out.println (t);

}

a b b c d e f a

Input -

Output -

Output - c - Output

~~Q8:~~

String s = "176 writes 34 4"

t = "176 3456 34"

string t = "",

t.toCharArray();

t = s.replace("writes", "3456")

\* Program to delete

String t = "1";

String s = "abcd1234 ASDF!@#";

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

{

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

{

}

else

{

t = t + s.charAt(i);

}

s.o.println(t);

O/P :- 1234ASDF!@#

### Program to delete lower case.

```

* string t = "";
string s = "abcHARI123$";
for (int i = 0; i < s.length(); i++)
{
    if (character.isLowerCase(s.charAt(i)))
    {
        t += s.charAt(i);
    }
}
s.o.println(t);

```

O/P:- HARI123\$

### Program to copy lower case -

```

string t = "";
string s = "abcH$";
for (int i = 0; i < s.length(); i++)
{
    if (character.isLowerCase(s.charAt(i)))
    {
        t += s.charAt(i);
    }
}
s.o.println(t);

```

O/P:- abc

\* This is my class — I/p  
0123456789101112131415  
is my class → O/p

### Substring

String s = "Today is my day";

```
int index = s.indexOf(' ');  
index++;  
System.out.println(s.substring(index));  
}
```

O/p → is my day

\* Program to replace part of a String.

```
StringBuffer sb1 = new StringBuffer("176 WATGR34");
```

```
StringBuffer sb2 = new StringBuffer("176 99934");
```

```
while (sb1.charAt(0) == sb2.charAt(0))
```

```
{
```

```
sb1.deleteCharAt(0);
```

```
sb2.deleteCharAt(0);
```

```
}
```

```
while (sb1.charAt(sb1.length() - 1) == sb2.charAt(sb2.length() - 1))
```

```
{
```

```
sb1.deleteCharAt(sb1.length() - 1);
```

```
sb2.deleteCharAt(sb2.length() - 1);
```

```
}
```

```
System.out.println(sb2);
```

O/p → 999

Replace character

[String Buffer]

```

StringBuffer sb1 = new StringBuffer("17G WATER 34");
sb1.setCharAt(0, '@');
sb1.toString();
    
```

Input

a b c d

1234

apple

Bananas

Banana

c b c d

3234

l P P l e

a a n a s

n a n a

\* class Replace23

{

    public static void main(String args[])

{

    StringBuffer sb1 = new StringBuffer("abcd");

    char c = sb1.charAt(sb1.length() - 2);

    sb1.setCharAt(0, c);

    System.out.println(sb1);

\* Program

Scanner scan = new Scanner(System.in);

String s = scan.nextLine();

String t = s.charAt(s.length() - 2) + s.substring(1);

System.out.println(t);

Input - APPLE

Output - lPPLe

## Program

Class Replace@

{

  Public String sum (String args[])

{

    Static void Scanner Scan = new Scanner (System.in);

    String S = Scan.nextLine();

    Change(S);

    Static void change (String S)

{

      StringBuffer sb = new StringBuffer (S);

      sb.setCharAt (0, sb.charAt (sb.length() - 2));

      S = S.println (sb);

}

Output: APPLE

APPLE

## Program to split a string

String S = "This is my day";

String ar[] = S.split (" ");

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

{

  S = O.println (ar[i]);

}

0	This
1	is
2	my
3	day

## Output:

This  
is  
my  
day.

## \* String - reverse

```
String s = "this is my day,";  
String ar[] = s.split(" ");  
for (int i = ar.length - 1; i >= 0; i--) {  
    System.out.println(ar[i]);  
}
```

Output  
day.  
my  
is  
This

## \* Program

Class Test

```
{  
    public static void main(String[] args) {  
        String s = "My name is ram.";  
        String ar[] = s.split(" ");  
        for (int i = 0; i < ar.length; i++) {  
            System.out.println(ar[i]);  
            Demo.change(ar[i]);  
        }  
    }  
}
```

O/P :-  
My  
name  
is  
ram.

## \* Program.

Class Demo

```
{  
    public static void main(String args...) {  
        String s = "This is my class";  
        String ar[] = s.split(" ");  
        for (int i = 0; i < ar.length; i++) {  
            rev(ar[i]);  
        }  
    }  
}
```

O/P :-  
int  
si  
ym  
ssalc

```
static void rev(String s)  
{  
    StringBuffer sb = new StringBuffer(s);  
    System.out.println(sb.reverse());  
}
```

Integer.parseInt()  $\Rightarrow$  requires String as an argument  
and that string must contain Integer characters  
only.

Program :- String to Integer

Class Test

{ public static void main (String args [])

{

String s = "44";

int n = Integer.parseInt (s);

s. o. pln (n+1);

}

O/P :- 45  
      

Program :- Anything to String

Double d = 4.4;

String s = String.valueOf (d);

s. o. pln (s);

O/P :- 4.4

Program :- Stack

import java. util. Stack;

Class Test

{

public static void main (String args [])

{

```

Stack stack = new Stack();
s.o.println(stack.isEmpty()); // true
s.o.println(stack.push('a'));// a
s.o.println(stack.push('b'));// b
s.o.println(stack); // [a, b]
s.o.println(stack.pop()); // b
s.o.println(stack); // [a]
s.o.println(stack.push(2)); // 2
s.o.println(stack.push(3)); // 3
s.o.println(stack); // [a, 2, 3]
s.o.println(stack.peek()); // 3
s.o.println(stack); // [a, 2, 3]
}
}

```

Program to reverse a string using Stack:

```

String s = "abcd";
String t = "";
Stack stack = new Stack(); // Stack <Character> stack = new Stack<Character>();
for (int i = 0; i < s.length(); i++)
{
    stack.push(s.charAt(i));
}
while (!stack.isEmpty() == false)
{
    t = t + stack.pop();
}
s.o.println(t);

```

abcd	0123
d c b a	_____

O/P :- dcba

## Conversion of Integer to Binary

```
int n = 5;  
String s = Integer.toBinaryString(5);  
System.out.println(s);
```

O/P :- 101

## Program :- Decimal to Octal

```
int n = 5;  
String s = Integer.toOctalString(n); // 13  
System.out.println(s); // O/P - 15
```

O/P :- 5

## Decimal to Octal

```
String s = Integer.toOctalString(13);  
System.out.println(s); // 15
```

## Decimal to Hexadecimal

```
String s = Integer.toHexString(15);  
System.out.println(s); // f
```

Decimal	Octal	Hex	Binary
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	10	8	1000
9	11	9	1001
10	12	a	1010
11	13	b	1011
12	14	c	1100
13	15	d	1101
14	16	e	1110
15	17	f	1111
16	20	10	10000
17	21	11	10001

\* Conversion Binary to Decimal

```
int n = Integer.parseInt("101", 2);
```

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

O/P :- 5

## Octal to Decimal

```
int n = Integer.parseInt("10", 8)
```

```
s.o.println(n);
```

O/p :- 14

## Program (Base 5)

```
int n = Integer.parseInt("11", 5);
```

```
s.o.println(n);
```

O/p :- 6

## Binary to Hex

```
String s = "101";
```

$B \rightarrow I$

```
int n = Integer.parseInt(s, 2);
```

$I - H$

```
String b = Integer.toHexString(n);
```

```
s.o.println(b);
```

O/p :- 5

## Octal to Hex

```
String s = "101";
```

$O \rightarrow I$

```
int n = Integer.parseInt(s, 8);
```

$I \rightarrow H$

```
String b = Integer.toHexString(n);
```

```
s.o.println(b);
```

O/p :- 5

## Decimal to Binary without Inbuilt

String t = " ";

Int n = 10;

while (n > 0)

{

    Int x = n % 2;

    t = x + t;

    n = n / 2;

}

s.o.pln(t);

O/P :- 1010

=

## Integer to Octal without Inbuilt :-

String t = " ";

Int n = 17;

while (n > 0)

{

    Int x = n % 8;

    t = x + t;

    n = n / 8;

}

s.o.pln(t);

O/P :- 21

\* Output of this

## Program :-

```

int res = 0;
int n = 1234;
while (n > 0)
{
    int x = n % 10;
    res = res * 10 + x;
    n = n / 10;
}
System.out.println(res);

```

O/P :- 4321

$$n = 1234 \quad 123 \quad 12 \quad 1 \times 0$$

$$x = 4 \quad 3 \quad 2 \quad 1$$

$$res = 0 \quad 4 \quad 43 \quad 432 \quad 4321$$

$$[res = 1]$$

while ( $n > 0$ )

$$\{ \quad n = n / 10;$$

$$x = n \% 10;$$

$$res = res * 10 + x;$$

$$\} \quad n = n / 2;$$

$$\frac{\text{reverse}}{10} = \frac{(0101)}{10}$$

$$\underline{10101} = 1010$$

## Program

```

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

```

String s = " THIS IS My CLASS. ";

String fh = " " + s.charAt(0);

String sh = s.substring(1);

fh = fh.toUpperCase();

sh = sh.toLowerCase();

System.out.println(fh + sh);

}

O/P :- This is my class.

1      2  
THIS IS my class.

\* Program to convert first letter of a String to upper case.

```
class Launch
{
    public static void main(String args[])
    {
        String s = "THIS IS My ClasS";
        String ar[] = s.split(" ");
        for (String data : ar)
        {
            String fh = " " + data.charAt(0);
            String sh = data.substring(1);
            fh = fh.toUpperCase();
            sh = sh.toLowerCase();
            System.out.println(fh + sh + " ");
        }
    }
}
```

Output This Is My Glass.

↑ Same output  
↓

\* Programs by calling method.

```
class Demo
{
    static void changeCase(String u)
    {
        String fh = u.substring(0, 1);
        String sh = u.substring(1);
        fh = fh.toUpperCase();
        sh = sh.toLowerCase();
        System.out.println(fh + sh + " ");
    }

    public static void main(String args[])
    {
        String s = "THIS IS My ClasS";
        String ar[] = s.split(" ");
        for (String data : ar)
        {
            changeCase(data);
        }
    }
}
```

## Program - 3

### class Demo

{

static String changeCase(String u)

{

String fh = u.substring(0, 1);

String sh = u.substring(1);

fh = fh.toUpperCase();

sh = sh.toLowerCase();

return fh + sh;

}

public static void main (String args [] )

{

String s = "This Is My class.";

String ar[] = s.split (" ");

for (String data : ar)

{

String r = changeCase (data);

s = s + r + " ";

// s = s + changeCase (data) + " ";

}

s . o . p ( );

{

Output This Is My Class.

Note :-

Below program is without any inbuilt methods.

\* Programs (Without built-in method)

class Demo

{

static String changeCase(String u)

{

String t = "";

if (u.charAt(0) >= 97 && u.charAt(0) <= 122)

{

t = t + (char)(u.charAt(0) + 32);

}

else

{

t = t + u.charAt(0);

}

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

{

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

{

t = t + (char)(u.charAt(i) + 32);

}

else

{

t = t + u.charAt(i);

}

}

return t;

public static void main(String args[])

{

\* main method is same as previous

Program \*

{

O/p :- This Is My Class.

## \* Methods.

Class Methods

{

s.o.p C String args[])

{

s.o.p (Math.sqrt(25)); // 5

s.o.p (Math.ceil(8)); // 2

s.o.p (Math.pow(2, 5)); // 32

s.o.p (Math.abs(-3.6)); // 3.6

s.o.p (Math.hypot(3, 4)); // 5

s.o.p (Math.max(5, 7)); // 7

s.o.p (Math.min(5, 7)); // 5

s.o.p (Math.round(5.499)); // 5

s.o.p (Math.floor(-0.499)); // -1

s.o.p (Math.ceil(-0.499)); // -0

s.o.p (Math.ceil(-1.0)); // -1

s.o.p (Math.random()); // 0.268987281 [0.0 to 1.0]

}

Program :-

\* s.o.p(m(50+int)(Math.random()\*10));

\* Program to create a random password.

String s = " abcdefghijklmnoprqstuvwxyz 1234567890\*~!@#\$%^&\*";

Scanner scan = new Scanner(System.in);

s.o.p (" Enter the length of the ~~pass~~ Password ");

int n = scan.nextInt();

String pswo = " ";

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

{

Put index = (int) (Math.random() \* 73);

psw = psw + s.charAt(index);

}

s.o.println(psw);

}

2

## ARRAYS

### Program

```

class Array
{
    public String args[])

    {
        //int ar[] = new int[5];
        (or)
        int ar[];
        ar = new int[5];
    }
}

```

### Program

```

class A2
{
    public String args[])
    {
        int ar[] = new int[5]; (or) int ar[] = {5, 4, 2, 3, 1};
        ar[0] = 5;
        ar[1] = 4;
        ar[2] = 2;
        ar[3] = 3;
        ar[4] = 1;

        for (int i = 0; i < ar.length; i++)
        {
            System.out.println(ar[i]);
        }
    }
}

O/P :- 5
       4
       2
       3
       1

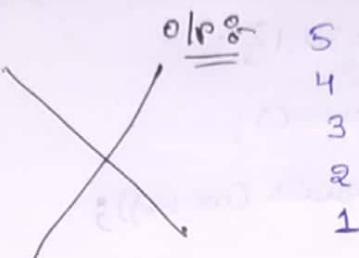
```

## Program to copy one array to another

```

int ar[] = {5,4,3,2,1};
int bx[] = ar;
for(int a : bx)
{
    System.out.println(a);
}

```



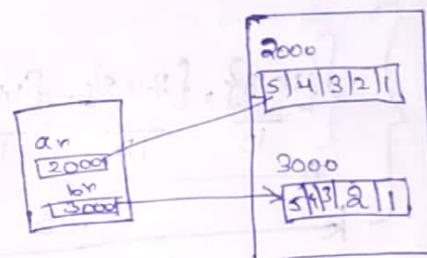
\* This is not the way to copy an array. \*

## Program :

```

int ar[] = {5,4,3,2,1};
int bx[] = new int[ar.length];
for (int i=0; i<ar.length; i++)
{
    bx[i] = ar[i];
}
for (int a : bx)
{
    System.out.println(a);
}

```

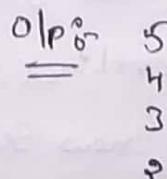


## Program

```

int ar[] = {5,4,3,2,1};
int bx[] = ar.clone();
for (int a : bx)
{
    System.out.println(a);
}

```

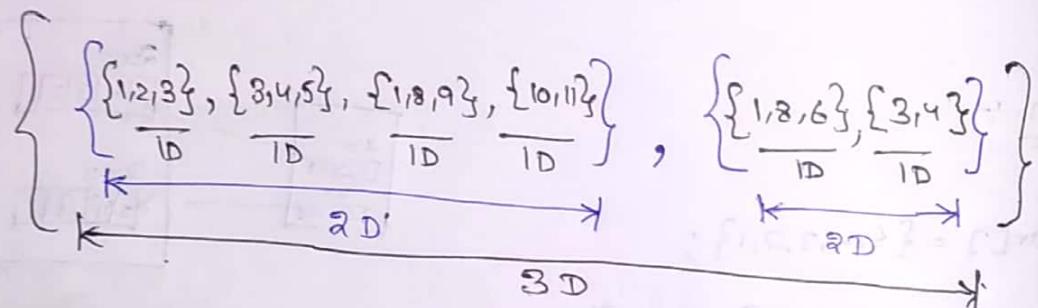


Program : inbuilt method to compare arrays.

```
int ar[] = { 5,4,3,2,1 };
int br[] = ar.clone();
System.out.println( Arrays.equals(ar,br));

```

O/P True



Program :- to compare 2D arrays. [deepEquals]

```
int ar[][] = {{5,4,3,2,1}, {5,8}};
int br[][] = {{5,4,3,2,1}, {5,8}};
System.out.println( Arrays.deepEquals(ar,br));

```

O/P True.

Program to copy word by word :-

```
Scanner sss = new Scanner(System.in);
String s= sss.nextLine();
Scanner scan = new Scanner(s);
while (scan.hasNext())
{
    System.out.println(scan.next());
}
```

Input:- My World  
Output:- My World

### Program to sort Array

```

import java.util.Arrays;
int arr[] = {5, 6, 2, 4, 3, 1, 7, 8, 9, 11};
Arrays.sort(arr);
for (int data : arr)
{
    System.out.print(data + " ");
}
System.out.println();

```

Output :- 1 2 3 4 5 6 7 8 9 11

### Program to sort section of Array

```

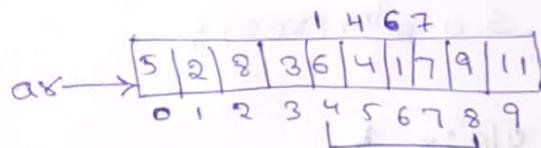
int arr[] = {5, 2, 8, 3, 6, 4, 1, 7, 9, 11};

```

```

Arrays.sort(arr, 4, 8);
for (int data : arr)
{
    System.out.print(data + " ");
}
System.out.println();

```



Output :- 5 2 8 3 4 6 7 9 11

### Program to copy section of Array

```

int arr[] = {5, 2, 8, 3, 6, 4, 1, 7, 9, 11};

```

```

int b[] = Arrays.copyOfRange(arr, 4, 8);
for (int data : b)
{
    System.out.print(data + " ");
}
System.out.println();

```

Output :- 6 4 1 7

### Program

```

int arr[] = {5, 4, 3, 2, 1} ;
Arrays.fill(arr, 4) ;
for (int data : arr)
{
    System.out.println(data + " ") ;
}
System.out.println();

```

O/p :- 4 4 4 4 4

### Program to find element is Present or Not:-

```
int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 11} ;
```

```
int res = Arrays.binarySearch(arr, 2) ; { returns index of
System.out.println(res); elements in array}
```

O/p :- 1

### Program to sort String

```
String arr[] = {"Hari", "Teja", "Vishat", "Kohli"} ;
```

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

```
for (String data : arr)
{
    System.out.println(data);
}
```

```
int index = Arrays.binarySearch(arr, "Ravi") ;
```

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

```
if (index >= 0)
```

```
{
    System.out.println("P");
}
```

```
else
```

```
{
    System.out.println("NP");
}
```

<u>O/p :-</u>	Hari	■
	Kohli	■
	Teja	-6
	Vishat	NP

String arr[] = {"12", "3", "1", "5", "9", "5"};

$$\text{Output} = 1 + 12 = 13.$$

String arr[] = {"12", "3", "1", "5", "5", "9"};

int n = Integer.parseInt(0);

int h = Integer.parseInt(2);

s.o.println(n+h);

Output :- 13

\* String arr[] = {"12", "3", "1", "5", "5", "9"};

int br[] = new int [arr.length];

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

    br[i] = Integer.parseInt(arr[i]);

}

Arrays.sort(br);

s.o.println(C br[0] + br[arr.length-1]);

Output :- 1 + 12

= 13

Swap :-

int a=5;

int b=6;

int temp=a;

a=b;

b=temp;

Initial	i
P	0
C	1
S	2
T	3

## Program to Sort C without built-in functions [BUBBLE SORT]

```

int arr[] = {5, 3, 2, 4, 1};
for (int i=0; i < arr.length-2; i++)
{
    for (int j=0; j < arr.length-2-i; j++)
    {
        if (arr[j] > arr[j+1])
        {
            int temp = arr[j+1];
            arr[j+1] = arr[j];
            arr[j] = temp;
        }
    }
    for (int data : arr)
    {
        System.out.print(data);
    }
    System.out.println();
}

```

Output :- 1 2 3 4 5

i	bubble
0	4
1	3
2	2
3	1

$$\begin{aligned}
 i + \text{bubble} &= 4 \\
 i + b &= \text{arr.length} - 1 \\
 b &\leq \text{arr.length} - 2 - i
 \end{aligned}$$

## Program [ Selection Sort ]

```
int arr[] = { 5, 3, 2, 4, 1 };
```

```
for (int i=0; i<arr.length-2; i++)
```

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

```
{
```

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

```
    { int t = arr[j];
```

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

```
        arr[i] = t;
```

```
}
```

```
for (int data : arr)
```

```
{
```

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

```
}
```

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

```
}
```

Output :- 1 2 3 4 5

→ Program      to find    max value

```
int arr[] = { 2, 1, 4, 3, 5, 6 };
```

```
int wife = arr[0];
```

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

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

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

```
}
```

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

arr → 

5	3	2	4	1
0	1	2	3	4

  
if j

\* with one iteration

we can sort smallest

element to the front. using Selection

sort.

(with a for loop)

Output :-

6

Program to find first and second largest number

```
int arr[] = {7, 1, 3, 5, 4, 6};  
int fwife = arr[0];  
int swife = 0;  
for (int i=0; i<arr.length; i++)  
{  
    if (arr[i] > fwife)  
    {  
        swife = fwife;  
        fwife = arr[i];  
    }  
    else if (arr[i] > swife)  
    {  
        swife = arr[i];  
    }  
}  
System.out.println(fwife);  
System.out.println(swife);
```

Output :- 7  
6

Program :-

```
int arr[] = {-7, 1, 1, -5, -4, 6, 0};  
int fwife = Integer.MIN_VALUE;  
int swife = Integer.MIN_VALUE;  
for (int i=0; i<arr.length; i++)  
{
```

```

if (CarCiJ > fwife)
{
    swife = fwife;
    fwife = arCiJ;
}

else if (CarCiJ > swife)
{
    if (CarCiJ != fwife)
    {
        s.wife = arCiJ;
    }
}

S.o.pln (fwife);
S.o.pln (swife);

```

Output :-      60  
                  1

Program      to find sum of 'n' numbers

```

int n = 4;
int sum=0;
for(int i=1; i<=n; i++)
{
    sum = sum + i;
}
S.o.pln (sum);

```

Output :-      10 //

## Factorial Program

```
int n=5;  
int fact=1;  
for(int i=0; i<n; i++)  
{  
    fact=fact*i;  
}  
s.o.pln(fact);
```

Output :- 120

## Program :-

```
int n=5000;
```

```
BigInteger fact = new BigInteger("1");
```

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

```
{ fact=fact.multiply(BigInteger.valueOf(i)); }
```

```
}
```

```
s.o.pln(fact);
```

Output :- 134-----0;

# Big Integers   Arithmetic   Methods

```
import java.util.BigInteger;

class Demo
{
    public static void main (String args[])
    {
        BigInteger a = new BigInteger("7");
        BigInteger b = new BigInteger("5");
        BigInteger res;
        res = a.add(b);
        System.out.println(res); // 12
        res = a.subtract(b);
        System.out.println(res); // 2
        res = a.mod(b);
        System.out.println(res); // 2
        int data = a.compareTo(b);
        System.out.println(data);
        res = a.divide(b);
        System.out.println(res); // 1
        res = a.multiply(b);
        System.out.println(res); // 35
    }
}
```

Program to find unit digit number

```
int x = 13456;
```

```
int y = 3;
```

```
x = x % 10;
```

```
BigInteger b = new BigInteger("1" + x);
```

```
s.o.p(b.pow(3).mod(BigInteger.valueOf(10)));
```

Output :- 6

Program :-

```
int n = 11;
```

```
BigInteger b = new BigInteger("4" + n);
```

```
if (b.isProbablePrime(1))
```

```
{
```

```
s.o.p("Prime");
```

```
}
```

```
else
```

```
{
```

```
s.o.p("not a prime");
```

```
}
```

→ 1 should be passed  
to this method  
to return a boolean  
value.

so → always returns  
a true.

Output :- Prime

\* Ternary Operators

```
datatype var = condition ? returns_if_true : returns_if_false;
```

```
String s = 5 > 2 ? "Yes" : "No";
```

Output = Yes.

## Program

```
int n=11;
```

```
BigInteger b = new BigInteger("11+n);
```

```
String res = b.isProbablePrime(1) ? "Prime" : "Not a Prime";
```

```
s.o.println(res);
```

Output :- Prime

## Program to print 50 to 100 prime numbers

```
int sn = 50;
```

```
int en = 100;
```

```
for(int i=sn+1; i<en; i++)
```

```
{
```

```
BigInteger b = new BigInteger("11+i);
```

```
if(b.isProbablePrime(1))
```

```
{
```

```
s.o.println(i);
```

```
}
```

Output :- 53, 59, 61, 67, --- 91, 97

## Program to print first 'n' prime numbers

```
int c=0; //count of Pno.
```

```
int n=3; //first 3 Pno.
```

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

```
{ BigInteger b = new BigInteger("11+i);
```

```
if(b.isProbablePrime(1))
```

```
{ s.o.println(i);
```

```
} c++;
```

Output :-

2

3

5

## Program to print 'n' prime numbers

```
Scanner Scan = new Scanner (System.in);
Scan.nextLine();
S.o.println ("How many prime no' you wanna print");
int n = Scan.nextInt();
S.o.println ("From where you wanna start ??");
int sn = Scan.nextInt();
int c = 0;
for (int i = sn; c < n; i++) {
    BigInteger b = new BigInteger ("4" + i);
    if (b.isProbablePrime (1)) {
        S.o.println (i);
        c++;
    }
}
```

## \* program

```
BigInteger b = new BigInteger ("2");
S.o.println (b.nextProbablePrime ());
```

Output :- 3

## Program      Conversion from String to character array

```
String s = "abcd";
char ar[] = new char[s.length()];
for(int i=0; i < s.length(); i++)
{
    ar[i] = s.charAt(i);
}
s.o.println(ar);
```

Output :-  
a  
b  
c  
d

## Program    using Inbuilt method

```
String s = "abcd";
char ar[] = s.toCharArray();
s.o.println(ar);
```

Output :-  
a  
b  
c  
d

## Program :-    Steps to sort a String:

- \* Convert String to char Array. // s.toCharArray();
- \* Sort Char Array. // Arrays.sort(ar);
- \* Convert Char Array to String. // String t = new String(ar);

## Program to sort String

```

String s = "bananas";
char arr[] = s. to Char Array ();
Arrays. sort (arr);
String t = new String (arr);
s. o. pln (t);

```

Output :- a a a b n n s

## Program to remove duplicates

```

String s = "bananas";
String t = "";
while (s.length() > 0)
{
    t = t + s. charAt (0);
    s = s. replace (" " + s. charAt (0), " ");
}
s. o. pln (t);

```

Output :- bans

## Program:- to know each character occurrence

```

String t = " abcadbace ";

```

```

char ar = t. to Char Array ();

```

```

Arrays. sort (ar);

```

```

String s = new String (ar);

```

// sorting string is completed.

```
while (s.length() > 0)
{
    char pc = s.charAt(0);
    int occ = s.lastIndexOf(pc) - s.indexOf(pc) + 1;
    s.out.println (pc + " - " + occ);
    s.replace (0+pc, " ");
}
```

Output &

a - 3

b - 2

c - 2

d - 1

e - 1 //

Write a Program to find the substring of 3 repeatedly

String s = "abcdefghijkl";

int k = 3;

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

{

s.o.println(s.substring(i, i + k));

}

Output :-

abc

bcd

cde

def

efg

fgh

ghi

hij

{ k.length() % 2 == 1 & & t.equals("+" + sb.reverse()) }

it gives the largest palindrome  
of odd numbers

Program to print largest palindrome

String s = "12112";

abc: for (int k = s.length(); k >= 1; k--)

{

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

{

String t = s.substring(i, i + k);

StringBuffer sb = new StringBuffer(t);

if (t.equals(sb.reverse() + "")),

{

s.o.println(t);

} break abc;

}

O/P :- 2112

## Program to Substring of Arrays

```
int arr[] = {0, 1, 2, 3, 4, 5, 6, 7};  
int k = 2;  
for (int i=0; i<arr.length-k; i++)  
{  
    for (int j=0; j<i+k; j++)  
    {  
        System.out.print(arr[j]);  
    }  
    System.out.println();  
}
```

Output:- 0123 1234 2345 3456 4567

## Program to find sum max of Array substring

```
int arr[] = {0, 1, 2, 3, 4, 5, 6, 7};  
int sum = 0;  
int max = Integer.MIN_VALUE;  
for (int k=1; k<=arr.length; k++)  
{  
    for (int i=0; i<=arr.length-k; i++)  
    {  
        for (int j=i; j<i+k; j++)  
        {  
            sum = sum + arr[j];  
        }  
        if (max < sum)  
        {  
            max = Math.max(max, sum);  
        }  
        sum = 0;  
    }  
}
```

S. O. P. M (max);

Output :- 28 //

```
if (sum > max)
{
    max = sum;
}
```

Program to left rotate the Array Once :

```
int ar[] = {1, 2, 3, 4};
int temp = ar[0];
for (int i=1; i<ar.length; i++)
{
    ar[i-1] = ar[i];
}
ar[ar.length-1] = temp;
for (int data: ar)
{
    S. O. P. M (data + " ");
}
```

Output:- 2 3 4 1

Write a program to left rotate array "n" number of times

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

```
int ar[] = {1, 2, 3, 4};
```

```
S. O. P. M ("Enter no. of left rotations");
```

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

```
n = n % ar.length;
```

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

{

```
    int temp = ar[0];
```

```
    for( int i=1; i < ar.length; i++)
```

{

```
        ar[i-1] = ar[i];
```

}

```
    int ar[ar.length-1] = temp;
```

}

```
for( int data:ar)
```

{

```
    System.out.println(data + " ")
```

}

Output :- 2

$n(20) \Rightarrow 20 / ar.length(4)$

3 4 1 2 //

1 2 3 4 //

Ho

5, 4, 3, 6, 1, 2, 9, 8, 11

Op 2, 4, 6, 8, 11, 9, 5, 3, 1

Ho// (4) 9, 11, (8) 6, 7, (12)  
      4, 6, 7, (8) 9, 11, (12)

$$\text{magic no} = 8G_1 - 8B$$

$$= 24 - 33$$

$$= -9 //$$

program

5	6	3	4	1	2	8
5	1	2	3	4	3	6

mul of even - sum of odd

$$48 \times 8 - 9$$

$$384 - 9$$

$$= \underline{\underline{375}}$$

\*

int arr[] = {5, 6, 3, 4, 1, 2, 8};

int mul = 9;

int sum = 0;

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

{

if (arr[i] % 2 == 0)

{

mul = mul \* arr[i];

}

else

{

sum = sum + arr[i];

}

System.out.println (mul - sum);

Output:-

375

## Program

```
int ar[] = {5, 6, 3, 4, 1, 2, 8};
```

```
int sume = 0;
```

```
int sumo = 0;
```

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

```
{
```

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

```
{
```

```
        sume += ar[i];
```

```
}
```

```
else
```

```
{
```

```
        sumo += ar[i];
```

```
}
```

```
}
```

```
if (sume > sumo)
```

```
{
```

```
s.o.println("Even wins");
```

```
}
```

```
else
```

```
{
```

```
s.o.println("Odd wins");
```

```
}
```

Output :-

===== Even wins //

### program

```
int arr[] = {5, 6, 3, 4, 1, 2, 8};
```

```
int sume = 0;
```

```
int sumo = 0;
```

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

```
{
```

```
    sume += arr[i];
```

```
}
```

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

```
{
```

```
    sumo += arr[i];
```

```
}
```

```
if (sume > sumo)
```

```
{
```

```
    s.o.pn ("even wins");
```

```
}
```

```
else
```

```
{
```

```
    s.o.pn ("odd wins");
```

```
}
```

Output :- Even wins.

### Program :-

arr

$$\begin{pmatrix} 1 & 2 & 3 \\ 00 & 01 & 02 \\ 4 & 5 & 6 \\ 10 & 11 & 12 \\ 7 & 8 & 9 \\ 20 & 21 & 22 \end{pmatrix}$$

br

$$\begin{pmatrix} 1 & 4 & 7 \\ 00 & 01 & 02 \\ 2 & 5 & 8 \\ 10 & 11 & 12 \\ 3 & 6 & 9 \\ 20 & 21 & 22 \end{pmatrix}$$

br	arr
0,0	0,0
0,1	1,0
0,2	2,0
1,0	0,1
1,1	1,1
1,2	2,1
2,0	1,0
2,1	2,2
2,2	2,2

## Program to transpose array

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

```
int br[3][3] = new int [3][3];
```

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

```
{
```

```
    for (int j=0; j<3; j++)
```

```
{
```

```
        br[i][j] = arr[i][j];
```

```
} s.o.::pmt();
```

```
for (int data: br) { i=0; j<3; j++)
```

```
{
```

```
    S.o.::cout << br[i][j] << " ";
```

```
} s.o.::pmt();
```

Output :- 
$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

## Program

arr

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

br      arr

0,0	2,0
0,1	1,0
0,2	0,0

1,0	2,1
1,1	1,1
1,2	0,1

2,0	2,2
2,1	1,2
2,2	0,2

Program to right rotate an array to 90°

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

int br[3][3] = new int [3][3];

for (int i=0; i<3; i++)

{

    for (int j=0; j<3; j++)

{

        br[i][j] = arr[3-i-1][j];

}

} ~~swap~~

for (int i=0; i<3; i++)

{

    cout << arr[i][i] + " "

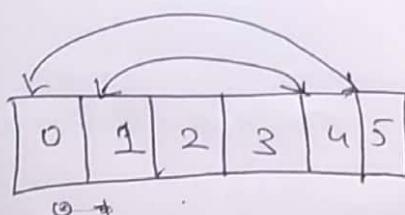
}

cout << endl;

Output

$$\begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

~~direct  
reverse~~



$$n-1-i$$

$$6-1-i$$

$$\frac{5-9}{0,5}$$

$$1,4$$

Mother method

for (int i=0; i<3; i++)

{ int k=2;

    for (int j=0; j<3; j++)

{

        br[i][j] = arr[k-1][j];

}

{

    cout << br[i][i] + " "

}

cout << endl;

1	2	3
4	5	6
7	8	9

Program

ar

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

br

$$\begin{bmatrix} 8 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$$

program:

to left rotate array

<u>by</u>	<u>ar</u>
0,0	0,2
0,1	1,2
0,2	2,2
1,0	0,1
1,1	1,1
1,2	2,1
2,0	0,0
2,1	1,0
2,2	2,0

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

int br[3][3] = new int [3][3];

for (int i=0; i<3; i++)

{  
~~K=0~~  
~~K=K+1~~

for (int j=0; j<3; j++)

{  
~~(j)~~  
br[i][j] = ar[~~K~~][n-1-i];  
}  
}

for (int i=0; i<3; i++)

{  
s.o.p (br[i][j]+ " ");  
}

s.o.pln();

- int k=0;

br[i][j] = ar[j][k];

- k--;

Output

$$\begin{bmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$$

## Program

```
int arr[] = { 5, 4, 3, 6, 1, 2, 9, 8, 11 };
int br[] = new int [arr.length];
arrays.sort(arr);
int ei = 0;
int op = arr.length - 1;
for (int i = 0; i < arr.length; i++) {
    if (arr[i] >= 0) {
        br[ei] = arr[i];
        ei++;
    } else {
        br[ei] = arr[i];
        ei--;
    }
}
System.out.println(br);
System.out.println();
```

## Output

2 4 6 8 11 9 5 3 1

## Program to sort without inbuilt method

```
Static int[] sortIt(int br[])
{
    for (int i=0; i<=br.length-2; i++)
    {
        for (int j=i+1; j<=br.length-1; j++)
        {
            if (br[i]>br[j])
            {
                int t = br[i];
                br[i] = br[j];
                br[j] = t;
            }
        }
    }
    return br;
}
```

```
Public static void main (String args[])
{
    int ar [] = {5,4,3,2,1};
    ar = sortIt(ar);

    for (int data:ar)
    {
        System.out.print(data + " ");
    }
    System.out.println();
}
```

Output : 1 2 3 4 5

## Program to merge two different arrays

```
static int[] merge(int ar[], int br[])
```

```
{
```

```
    int cr[] = new int [ar.length + br.length];
```

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

```
{
```

```
        cr[i] = ar[i];
```

```
}
```

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

```
{
```

```
        cr[ar.length+i] = br[i];
```

```
}
```

```
    return cr;
```

```
}
```

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

```
{
```

```
    int a[] = {1, 2, 3, 4};
```

```
    int b[] = {5, 6, 7, 8, 9, 10};
```

```
    int c[] = merge(a, b);
```

```
    for (int data : c)
```

```
{
```

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

```
}
```

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

```
}
```

Output: 1 2 3 4 5 6 7 8 9 10

## Program

```
String S = "BCD";
```

```
char C = (char)66;
```

```
S.o.println(S.contains(C+"u));
```

Output :- true.

## Panagram    Program    (or)    Not    Panagram

```
String S = "The quick brown fox jumps over a lazy dog";
```

```
S = S.toUpperCase();
```

```
int C=0;
```

```
for (int i=65; i<=90; i++)
```

```
{
```

```
    char PC = (char)(i);
```

```
    if (S.contains(C+"u+PC))
```

```
{
```

```
        C++;

```

```
}
```

```
}
```

```
if (C==26)
```

```
{
```

```
    S.o.println("Panagram");
```

```
}
```

```
else
```

```
{
```

```
    S.o.println("Not a Panagram");
```

```
}
```

Output :- Panagram

## Anagram

- \* Remove Spaces
- ① Convert to lowercase (or) uppercase
- ② Convert String to array
- ③ Sort the arrays
- ④ Compare using Arrays.equals

## Program

```
String s1 = "Mother in Law";
```

```
String s2 = "Hitler Woman";
```

```
s1 = s1.replace(" ", "");
```

```
s2 = s2.replace(" ", "");
```

```
s1 = s1.toLowerCase();
```

```
s2 = s2.toLowerCase();
```

```
char ar1[] = s1.toCharArray();
```

```
char ar2[] = s2.toCharArray();
```

```
Arrays.sort(ar1);
```

```
Arrays.sort(ar2);
```

```
if (Arrays.equals(ar1, ar2))
```

```
{
```

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

```
}
```

```
else
```

```
{
```

```
System.out.println("Not a Anagram");
```

```
}
```

Output: Anagram

Program to find occurrence of each alphabet

String s = "banana\$";

s = s. toUpper Case();

int ar[] = new int [26];

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

{

    int index = s.charAt(i) - 65;

    ar[index] = ar[index] + 1;

}

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

{

    if (ar[i] > 0) // → Remove to print occurrence of each character

{

        System.out.println((char)(65+i) + " --- " + ar[i]);

}

    System.out.println();

A	B
3	1
0	25

Output:-

A --- 3

B --- 1

N --- 2

S --- 1

## Program to split string without built-in method

String s= "This is my class.";

int sc=0;

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

{

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

{

sc++;

}

int wc= sc+1;

String ar[] = new String [wc];

int h=0;

String word= "";

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

{

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

{

ar[h] = word;

h++;

} word= "";

else

{

word = word + s.charAt(i);

} }

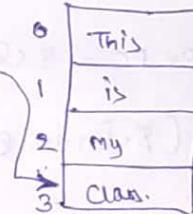
ar[h] = word;

for( String data: ar)

{

s.o.println(data);

}



### Output

This

is

my

class.

## Program to print unique character in a string

```
String s = "SILLY SPIDGRS";
s = s.replace(" ", "");
String t = "";
for (int i=0; i < s.length(); i++) {
    char pc = s.charAt(i);
    if (s.indexOf(pc) == s.lastIndexOf(pc))
        t = t + (pc + " ");
}
System.out.print(t);
```

Output:- YPDER

## Program to remove spaces from starting & ending

```
String s = "India - is ... great. --- ";
String t = "";
int si = 0, ei = 0;
for (int i=0; i < s.length(); i++) {
    if (s.charAt(i) == ' ')
        ;
    else
        {
            si = i;
            break;
        }
}
```

```

for (int i=0; i < s.length(); i++)
{
    if (s.charAt(i) == ' ')
    {
        i++;
    }
    else
    {
        c[i] = s[i];
    }
}

```

```

for (int i=s.length(); i >= 0; i--)
{
    t = t + s.charAt(i);
    s.o.println(t);
}

```

Output :- India is great.

Program to remove extra spaces Pro b/w

String s = "This ---- is my --- class."

```

for (int i=0; i < s.length(); i++)
{
    if (s.charAt(i) == ' ' && s.charAt(i+1) == ' ')
    {
        i++;
    }
    else
    {
        t = t + s.charAt(i);
    }
}
s.o.println(t);

```

Output :- This\_is\_my\_class.

HW

$a_2 c_3 //$

program to add numbers in the string.

String s = "123456";

int sum = 0;

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

{  
if (s.charAt(i) >='0' & & s.charAt(i) <='9') // character is Digit (charAt ci)

{

sum = sum + (s.charAt(i) - 48);

} // Integer.parseInt (s.charAt(i));

s.o.println (sum);

Output :- 1+2+3+4+5+6 = 21

Program

String s = "123456";

int sum = 0;

String t = "";

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

{

if (Character.isDigit (s.charAt(i)))

{

sum = sum + Integer.parseInt ("+" + s.charAt(i));

}

else

{

t = t + s.charAt(i);

}

s.o.println (sum + t);

Output :- 123456

## Program

```
String s= "12345";
int c=0;
for(int i=0; i<s.length(); i++)
{
    if (s.charAt(i)>='0' && s.charAt(i)<='9')
    {
        c++;
    }
    else
    {
        c++;
    }
}
if (c==0)
{
    System.out.println("Number");
}
else
{
    System.out.println("Not a Number");
}
```

Output :- Number.

## Program

```
String s= "Ta87his i35 my cl394lass.";
```

```
String ar[] = s.split(" ");
```

```
for(String d:ar)
```

```
{  
    add(d);  
}
```

```

s.o.println();
}

static void add (String s)
{
    int sum=0;
    String t= "";
    for (int i=0; i<s.length(); i++)
    {
        if (s.charAt(i)>='0' && s.charAt(i)<='5')
        {
            sum= sum + s.charAt(i)-48;
        }
        else
        {
            t=t+s.charAt(i);
        }
    }
    s.o.println (sum+t+" ");
}

```

T9h8i7s	133	m4y	1c31as24s.
0	1	2	3

Output :- 24 This 3 is my 10 class.

## Program

```
String s = "aB1cD2";
String t = "";
if (s.charAt(0) >= 97 && s.charAt(0) <= 122)
{
    t = t + (char)(s.charAt(0)-32);
}
else
{
    t = t + s.charAt(0);
}

for (int i=1; i < s.length(); i++)
{
    if (s.charAt(i) >= 65 && s.charAt(i) <= 90)
    {
        t = t + (char)(s.charAt(i)+32);
    }
    else
    {
        t = t + s.charAt(i);
    }
}
s.o.println(t);
```

Output :- Ab1cd2

HQ

```
for (int i=0; i < s.length(); i++)
{
    if (s.charAt(i) == 'a' && s.charAt(i) == 'A')
    {
        int n = s.charAt(i) - 48;
        for (j=0; j < n; j++)
        {
            s[i].print(s.charAt(i-1));
        }
    }
}
```

a2b4c1

## Program

```

Scanner scan = new Scanner (System.in);

System.out.println("User Enter the age");
int age = scan.nextInt();

System.out.println("User Enter the name");
scan.nextLine();

String name = scan.nextLine();

System.out.println("age is "+ age);
System.out.println("name is "+ name);

```

Output :- 23  
dheeraj

## Program

```

String s = "aabccccceee";
int c = 1;
for (int i=0; i < s.length() - 1; i++) {
    if (s.charAt(i) == s.charAt(i+1)) {
        c++;
    } else {
        System.out.println(s.charAt(i) + " -- " + c);
        c = 1;
    }
}
System.out.println(s.charAt(s.length() - 1) + " -- " + c);

```

a a b c c c c e e e  
 0 1 2 3 4 5 6 7 8 9  
 ↑↑↑↑↑↑↑↑↑↑↑↑↑↑  
 s.length() - 1

Output :- a -- 2  
b -- 1  
c -- 4  
e -- 3

## Program

a b b b c d d d d d e e f g g g g g  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
 ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

c = 1;  
 =

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

c = x 2 3

a - 1

c = x 2 3 4 5

b - 3

c = x 2

c - 1

c = 1

d - 5

c = x 2 3 4 5

e - 2

f - 1

g - 5

Program to highest occurrence

String s = "a b b b e d d d d d e e f g g g g g";

int c = 1;

int max = 0;

char gc = ' ';

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

{

if (s.charAt(i) == s.charAt(i + 1))

{

  c++;

}

else

{

  if (c >= max)

{

    max = c;

    gc = s.charAt(i);

}

  c = 1;

}

  if (c >= max)

  {

    max = c;

    gc = s.charAt(i);

}

  c = 1;

s.out.println(gc + " " + max);

Output :-     g - 5

Program to display arrays  
String

```
int arr[] = {1, 4, 5};
```

```
s.o.::pLn ( Arrays::to String (arr) );
```

```
int arr[2][2] = {{1, 4, 5}, {6, 7, 8}};
```

```
s.o.::pLn ( Arrays::deepToString (arr) );
```

Output

[1, 4, 5]

[1, 4, 5], [6, 7, 8]

Program

n=6

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

```
{
```

```
    int c=0;
```

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

```
{
```

```
        res = arr[i]*k+c;
```

```
        arr[i] = res % 10;
```

```
c = res / 10;
```

```
} }
```

## Program to find next Palindrome number

```
import java.util.*;  
class Solution  
{  
    public static void main (String args[])  
    {  
        Scanner scan = new Scanner (System.in);  
        int n = scan.nextInt();  
        for (int i=n+1; ; i++)  
        {  
            if (isPal(i))  
            {  
                System.out.println(i);  
                break;  
            }  
        }  
  
        static boolean isPal (int n)  
        {  
            String s = "" + n;  
            StringBuffer sb = new StringBuffer (s);  
            return s.equals ("+" + sb.reverse());  
        }  
}
```

Output :- Sample I/p - 129

Output - 131.

Program to find      nearest number      which contains all nos.

Program -1

class Solutions

{

    public static void main (String args [] )

{

        Scanner scan = new Scanner (System. in );

        int n = scan.nextInt ();

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

{

    if ( isAnagram (i))

{

        System.out.println (i);

        break;

}

}

    static boolean isAnagram (int i) {

{

        String s1 = "" + i;

        String s2 = "" + i;

        char ar[] = ~~int~~ s1.toCharArray ();

        char br[] = s2.toCharArray ();

        Arrays.sort (ar);

        Arrays.sort (br);

        return Arrays.equals (ar, br);

}

}

Output :-

1234 - Sample Input

1243 - Output

## Program - 2

psvm CString args[])

{

int n = 123;

for (int i = n; i < 1000; i++)

{

if (isAnagram(i, n))

{

SOP(i);

}

}

## Output

123

132

213

231

312

321

## Program

psvm CString args[])

{ String s = "abcd";

int n = 1234;

for (int i = n; i < 10000; i++)

{

if (isAnagram(i, n))

{

Print(i);

}

}

}

Print static void print (int i, String s)

{

String sum = " " + i;

String t = " "

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

{

int index = (s.charAt(j) - 48) - 1;

t = t + s.charAt(index);

}

S.O. p.lm(t);

}

### Program

### [Recursion]

Class Solution

{

P.S.V.m (String args[])

{

    incx(60);

}

    Static void incx(int n)

{

        S.O.p.lm(n);

}

}

Output :- 60.

### Program - recursion

P.S.V.m (String args[])

{

    incr(0);

}

    Static void incr(int n)

{

        if (n <= 2)

            S.O.p.lm();

        incr(n+1);

}

}

}

Output :- 0 1 2

incr(3)	n	3
if (n <= 2)	n	3
S.O.p.lm();		X
incr(n+1);		
incr(2)	n	2
if (n <= 2)	n	2
S.O.p.lm();		
incr(n+1);		
incr(1)	n	1
if (n <= 2)	n	1
S.O.p.lm();		
incr(n+1);		
incr(0)	n	0
if (n <= 2)	n	0
S.O.p.lm();		
incr(n+1);		
main()		
incr(0);		

## Program

Class Solution

{

PSUMC(String args[])

{

INCR(3);

{

Static void INCR(int n)

{

If(n >= 0)

{

SOP(n);

INCR(n-1);

}

}

Output:-

3

2

1

0

## Reverse string using Recursion

Class Solutions

{

PSUMC(String args[])

{

String s = "abcd";

INCR(3, s)

SOP(s);

}

Static void INCR(int n, String s)

{

If(n >= 0)

{

SOP(s.charAt(n));

INCR(n-1, s);

}

}

}

## Program

```
psum (String args[])
```

```
{
```

```
String s = "01234567";
```

```
String res = add(s);
```

```
s.o.println(res);
```

```
}
```

```
static String add (String s)
```

```
{
```

```
if (s.length() == 1)
```

```
return s;
```

```
return add (s.substring(1)) + s.charAt(0);
```

```
}
```

## factorial using Recursion

```
psum (String args[])
```

```
{
```

```
int n = 3;
```

```
int res = fact(n);
```

```
s.o.println(res);
```

```
}
```

```
static int fact (int n)
```

```
{
```

```
if (n == 0)
```

```
return 1;
```

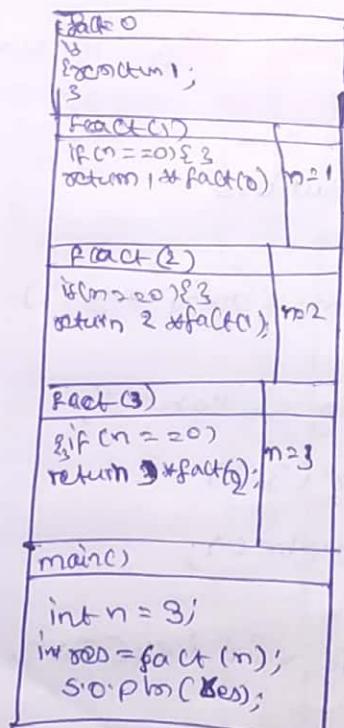
```
return n * fact(n-1);
```

```
}
```

Output :- 6 //

## Output

76543210



## Program to add numbers

psvm (String args[])

{

```
int n = 5231;  
int res = add(n);  
System.out.println(res);
```

}

static int add(int n)

{

```
if (n == 0)  
{  
    return 0;  
}
```

}

return (n % 10) + add(n / 10);

Output = 11 [5+2+3+1]

## Spiral

## Pattern

```
import java.util.*;
```

class Solution

{

```
Static int n = 4, x = 0, c = 0, data = 1;
```

```
Static int ar[][] = new int[n][n];
```

Static void right()

{

```
If (x >= 0 && c <= 0 && x < n && c > n && ar[x][c] == 0)
```

{

```
ar[x][c] = data;
```

```
data++;
```

```
c++;
```

} right();

```
static void down()
{
    if(x>=off & c>=off & x<n + & c<n & ar[x][c]==0)
    {
        ar[x][c]=data;
        data++;
        x++;
        down();
    }
}

static void left()
{
    if(r>=off & c>=0 & x<n & c<n & ar[x][c]==20)
    {
        ar[x][c]=data;
        data++;
        c--;
        left();
    }
}

static void up()
{
    if(x>=off & c>=off & x<n & c<n & ar[x][c]==20)
    {
        ar[x][c]=data;
        data++;
        x--;
        up();
    }
}
```

```

public static void main (String args[])
{
    while (data <= n*n)
    {
        right();
        c-- ; x++;
        down();
        c-- ; x--;
        left();
        c++ ; x--;
        up();
        c++ ; x++;
    }
    for (int i=0; i<n; i++)
    {
        for (j=0; j j<n; j++)
        {
            System.out.printf ("%02d", arr[i][j]);
        }
        System.out.println();
    }
}

```

Output :-

01	02	03	04
12	13	14	05
11	16	15	06
10	09	08	07

### Spiral - 2

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

$$x = n - 1; c = n - 1$$

while (data <= n \* n)

{

    up();

    c--; x++;

    left();

    c++; x++;

    down();

    c++; x--;

    right();

    c--; x--;

}

### Spiral - 3

while (data < n \* n)

{

    up();

    x++; c++;

    right();

    x++; c++;

    down();

    c--; x--;

    left();

    x--; c++;

4 5 6 7

3 14 15 8

2 13 16 9

1 12 11 10

$$x = n - 1; c = 0$$

## Spiral - 4

13	14	15	16
12	3	21	5
11	2	1	6
10	9	8	7

```
while ( data < n * n )
```

۲

left( ).

C++; & ++;

down ( );

X--; C++;

```
right());
```

C - - ; & - - ;

up to;

`++;` `--;`

۳

Spiral-5

16	19	14	13
5	4	3	12
6	1	2	11
7	8	9	10

		4	3	2	1
		5	14	13	12
		6	15	16	11
		7	8	9	10
					<u> </u>

$$c = n - 1 ; \quad x = 0$$

$$S.O.P \left( \frac{1}{1.02d} ", (n \times n + 1) - \alpha(n) \right)$$

12 34

12 13 14 5

12 13 14 5  
v 16 15 6

U 16 6  
10 9 8 7

18987  
was made and

$$SOP \left( "1.02d", [n+n+1] - av [r]cc \right);$$

## RAT Program

	0	1	2	3	4
0	↑↓				
1	→↑↓	→↓	→↓	→↓	
2	↓↑				
3	↓↑				
4	→↑	→↑	→↑	→↑	↓↑
5				↑↑	→↑

class Demo

{

static int n = 4;

static int ar[5][5] = { {1, 0, 0, 1, 0},  
{1, 1, 1, 0, 0},  
{0, 1, 1, 0, 0},  
{0, 0, 1, 1, 0} };

public static void main (String args[]).

{ Boolean res = move(0, 0);  
System.out.println(res);  
}

static boolean move (int x, int c)

{

if (x == 0 && c == 0 && x < n && c < n)

{

if (x == n - 1 && c == n - 1 && ar[x][c] == 1)

{

return true;

```

3
else if (arr[x][c] == 1)
{
    if (move(x, c+1))
    {
        return true;
    }
    if (move(x+1, c))
    {
        return true;
    }
    return false;
}
else
{
    return false;
}
else
{
    return false;
}
}

```

Output :-

false

H -> Swap

## No. of ways to reach final point

class Demo

{

    Static int n=4, count;

    Static int arr[4][4] = { {1,1,0,1},

          {1,1,1,0},

          {0,1,1,0},

          {0,0,1,1} };

    Public static void main (String args[])

{

        move (0,0);

        S.O.Pm (Count);

}

    static void move (int x, int c)

{

        if (x>=0 && c>=0 && x<n && c<n)

{

          if (x==n-1 && c==n-1 && arr[x][c]==1)

{

          Count++;

}

        else if (arr[x][c]==1)

{

          move (x,c+1);

          move (x+1,c);

} }

        Output :- 4 //

{

Write a program to print first 'n' Fibonacci series

class Demo

```
{ public String args[])
```

```
{ int n=2;
```

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

```
    if (n>1)
```

```
    { ar[1]=1;
```

```
}
```

```
    for (int i=2; i<ar.length; i++)
```

```
{
```

```
    ar[i] = ar[i-1] + ar[i-2];
```

```
}
```

```
    for (int data : ar)
```

```
{
```

```
    S.O.Println(data);
```

```
}
```

```
}
```

Output

0  
1

Program to print nth element in Fibonacci series

```
int n=6;
```

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

```
if (n>1)
```

```
{ ar[1]=1;
```

```
}
```

```
for (int i=2; i<ar.length; i++)
```

```
{ ar[i] = ar[i-1] + ar[i-2];
```

```
}
```

```
S.O.Println(ar[n-1]);
```

Output

5 //

## Program to find nth element in Fibonacci using recursion

Class Demo

{

    PSVM CString args[])

{

    int res = fib(6);

    S.O.Pln(res);

static int fib(int n)

{

    if (n == 1)

{

        return 0;

}

    if (n == 2)

{

        return 1;

}

    return fib(n-1) + fib(n-2);

{

}

Output = 5

Program to find nth element in Fibonacci using recursion

Class Demo

```
{  
    p s v m CString args[])
```

```
{
```

```
    int res = fib(6);
```

```
    S.0. Pbm(res);  
}
```

```
static int fib (int n)
```

```
{  
    if (n == 1)
```

```
{
```

```
        return 0;
```

```
}
```

```
    if (n == 2)
```

```
{
```

```
        return 1;
```

```
}
```

```
    return fib (n-1) + fib (n-2);
```

```
}
```

```
{
```

Output = 5

Program to find factors

```
int n = 10;
```

```
int c = 0;
```

```
S.0. Pbm (1);
```

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

{

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

{

s.o.plm(i);

{

}

s.o.plm(n);

Output :-

1

2

5

10

\* Program to find prime no.

```

int n=5;
int c=0;
for (int i=2; i<=n/2; i++)
{
    if (n % i == 0)
    {
        c++;
    }
}
if (c==0)
{
    s.o.plm("Prime");
}
else
{
    s.o.plm ("! a Prime");
}

```

Output

Prime

## Program

Put  $n = 5;$

int boolean c = false;

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

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

{ c = true;  
break;

}

if (c == false)

{ System.out.println ("Prime");

}

else

{ System.out.println ("Not prime");

}

Output :- Prime.

## program

int n = 6;

int m = 15;

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

{ if ( $m \% i == 0$  &  $m \% i == 0$ )

System.out.println (i);

}

Output :- 1, 3.

## Program of GCD

```
int n=6;  
int m=15;  
for (int p = Math.min(n,m); p >= 1; p--) {  
    if (n % p == 0 & m % p == 0)  
        break;
```

```
}
```

Output :-

## Program

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

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

```
{
```

```
    if (i == j)
```

```
.Continue;
```

```
    int val = ar[i] + 10 + ar[j];
```

```
BigInteger b = new BigInteger (" "+val);
```

```
if (b . is probable prime(1))
```

```
{ S.O.P(val);
```

```
}
```

```
} }
```

## Output

17, 71, 73, -

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

11, 22, 33, 44, 55, ---- 99.

### Program

```
class Dog  
{  
    static Dog d;
```

```
private Dog ()
```

```
{  
    System.out.println ("Dog has been created");  
}
```

```
static Dog create()
```

```
{  
    if (d == null)
```

```
{  
    d = new Dog();  
}
```

```
}  
return d;  
}
```

```
}  
class Test
```

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

```
{  
    Dog d1 = Dog.create();  
    Dog d2 = Dog.create();  
}
```

```
}
```

## Program to find Strong Number

```

int n=145;
String s = "" + n;
int sum = 0;
for (int i=0; i < s.length(); i++) {
    sum = sum + fact(s.charAt(i) - 48);
}

```

if (sum == n)

```

    s.o.pln("Strong");
}

```

else

```

    s.o.p("! Strong");
}

```

Output :- Strong //

## Program to find Armstrong numbers

```

int n=153;

```

```

String s = "" + n;

```

```

int d = s.length();

```

```

for (int i=0; i < s.length(); i++) {
    sum = sum + Math.pow(s.charAt(i) - 48, d);
}

```

if (sum == n)

```

    s.o.p("Armstrong");
}

```

```

else
    s.o.p("! a Armstrong");
}

```

$$\begin{aligned}
 145 &\leftarrow \\
 1! + 4! + 5! & \\
 1 + 24 + 120 = 145
 \end{aligned}$$

153 - Armstrong

$$3^3 + 5^3 + 3^3 = 153$$

8208 - Armstrong

3 →  
No. of  
digits.

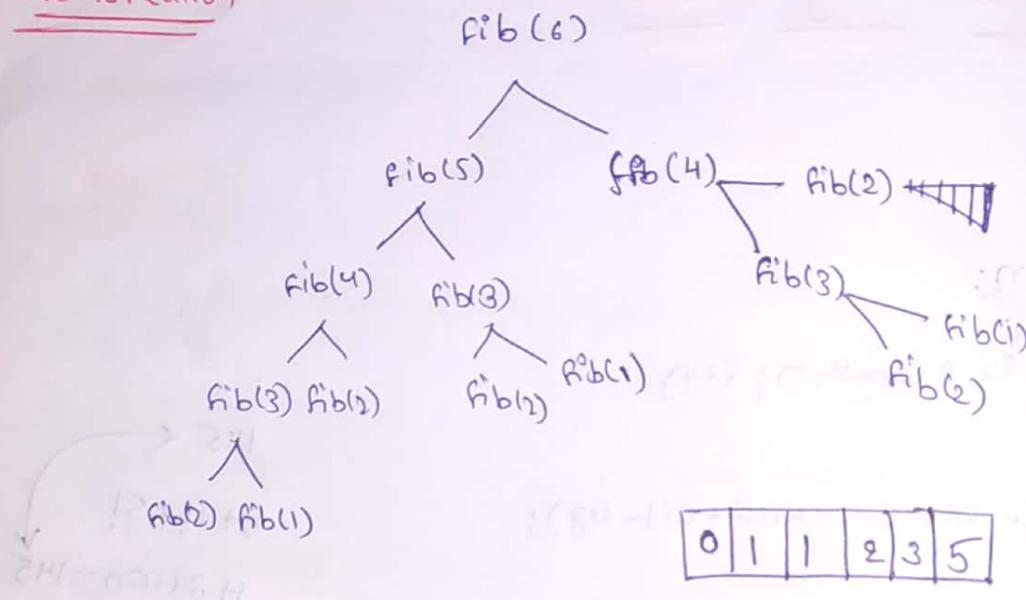
Output //

153 - Armstrong

$$3^3 + 5^3 + 3^3 = 153$$

8208 - Armstrong

## Memoization



## Program

1 2 3 4 5 6 7 8 9 10	1 3 2 6 5 4 10 9 8 7	<u>Output</u> $\swarrow$
-------------------------------	-------------------------------	-----------------------------

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

```
{
```

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

```
{
```

```
    if (j == i)
```

```
{
```

```
      t = (c++ + " ") + t;
```

```
{
```

```
s.o.p(t);
```

```
t = " ";
```

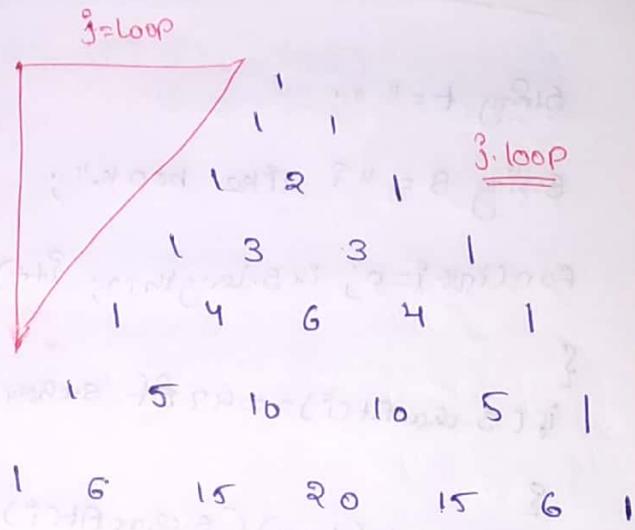
```
{
```

```
"
```

## Program

one loop

1									
1	1								
1	2	1							
1	3	3	1						
1	4	6	4	1					
1	5	10	10	5	1				
1	6	15	20	15	6	1			



\* Fact ( $i$ ) /  $c \cdot \text{fact}(i-j) * \text{fact}(j)) + " "$  );

$$n_{cr} = \frac{n!}{(n-r)! \cdot r!}$$

$$\frac{i!}{(i-j)! \cdot j!} = {}^i C_j$$

## program

```

for (int i=1; i<=9; i++)
{
    for (int j=1; j<=9; j++)
    {
        if (j==i)
        {
            continue;
        }

        for (int k=1; k<=9; k++)
        {
            if (k==i || k==j) continue;

            if ((i*10+j) + (j*10+k) + (k*10+i) == (i*100+j*10
                +k))
            {
                System.out.println(i+j+k);
            }
        }
    }
}

```

$$\begin{array}{c}
 11 + 99 + 88 \\
 \backslash \quad | \quad /
 \\ 198
 \end{array}$$

$$\begin{array}{c}
 11 + 22 + 33 \\
 \backslash \quad | \quad /
 \\ 123
 \end{array}$$

$$i=1 \\ j=12$$

$$k=123$$

## Encrypt - Program

String t = " ";

String s = "I like book";

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

{

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

{

t = t + (char)(s.charAt(i)+k);

}

else

{

k = t+s.charAt(i);

}

}

s.o.println(t);

### Output :-

i like book  
k nkmg dnm

### Program

int k=500;

~~int~~ k = k % 26;

String t = " ";

String s = "afbB123#\_24";

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	10010	10011	10012	10013	10014	10015	10016	10017	10018	10019	10020	10021	10022	10023	10024	10025	10026	10027	10028	10029	10030	10031	10032	10033	10034	10035	10036	10037	10038	10039	10040	10041	10042	10043	10044	10045	10046	10047	10048	10049	10050	10051	10052	10053	10054	10055	10056	10057	10058	10059	10060	10061	10062	10063	10064	10065	10066	10067	10068	10069	10070	10071	10072	10073	10074	10075	10076	10077	10078	10079	10080	10081	10082	10083	10084	10085	10086	10087	10088	10089	10090	10091	10092	10093	10094	10095	10096	10097	10098	10099	100100	100101	100102	100103	100104	100105	100106	100107	100108	100109	100110	100111	100112	100113	100114	100115	100116	100117	100118	100119	100120	100121	100122	100123	100124	100125	100126	100127	100128	100129	100130	100131	100132	100133	100134	100135	100136	100137	100138	100139	100140	100141	100142	100143	100144	100145	100146	100147	100148	100149	100150	100151	100152	100153	100154	100155	100156	100157	100158	100159	100160	100161	100162	100163	100164	100165	100166	100167	100168	100169	100170	100171	100172	100173	100174	100175	100176	100177	100178	100179	100180	100181	100182	100183	100184	100185	100186	100187	100188	100189	100190	100191	100192	100193	100194	100195	100196	100197	100198	100199	100200	100201	100202	100203	100204	100205	100206	100207	100208	100209	100210	100211	100212	100213	100214	100215	100216	100217	100218	100219	100220	100221	100222	100223	100224	100225	100226</td

```

for (int i=0; i<s.length(); i++)
{
    if (s.charAt(i) >= 97 && s.charAt(i) <= 122)
        t = t + (char)(s.charAt(i) + k);
    else if (s.charAt(i) + k >= 123)
        t = t + (char)(s.charAt(i) + k - 26);
    else if (s.charAt(i) >= 65 && s.charAt(i) <= 90)
        t = t + (char)(s.charAt(i) + k);
    else
        t = t + (char)(s.charAt(i) + k - 26);
}
s.append(t);

```

## Command Line Arguments

C:\Users\ABC>d:

D:\>cd P

D:\P>javac Demo018.java

D:\P>java Demo018

hello

### Program:-1

```
class Demo018
```

```
{
```

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

```
{
```

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

```
}
```

### Program-2

```
class Demo018
```

```
{
```

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

```
{
```

```
    System.out.println(args[0]);
```

```
    System.out.println(args[1]);
```

```
}
```

```
}
```

\*

D:\P>java Demo018

123

abc

123

abc

### Program-3

```

class Demo18

{
    public static void main(String ... args)
    {
        for (int i=0; i<args.length; i++)
        {
            System.out.println(args[i]);
        }
    }
}

```

### CMD

D:\> java Demo18      123 abc xyz

123

a bc

xyz

### Program-4

class Demo18

```

{
    public static void main(String ... args)
    {
        int f = Integer.parseInt(args[0]);
        int s = Integer.parseInt(args[1]);

        System.out.println(f+s);      //      System.out.println((f+s)/e);

        System.out.println((f+s)/e);
    }
}

```

### CMD      Addition

D:\> java Demo18 5 6      //

//

### Division

D:\> java Demo18

5 6

5 //

## Program - 5

## Area of circle

Class Demo18

```
{
    public static void main(String[] args) {
        int r = Integer.parseInt(args[0]);
        System.out.println(2 * Math.PI * r);
    }
}
```

## cmd

D:\p> java Demo18 8

50.26548245743669

## Program 6

## Area of triangle

Class Demo18

```
{
    public static void main(String[] args) {
        int b = Integer.parseInt(args[0]);
        int h = Integer.parseInt(args[1]);
        System.out.println((b * (float) h) / 2);
    }
}
```

## cmd

D:\p> java Demo18 5 7

17.5

### Program - 7

Average of all arguments.

```

class Demo18
{
    public static void main (String args[])
    {
        int sum = 0;
        for (int i=0; i<args.length; i++)
        {
            sum = sum + Integer.parseInt(args[i]);
        }
        System.out.println (float) sum / args.length;
    }
}

```

### Cmd

D:\> java Demo18 1 2 3 4 5 6 7 8 9  
5.0

### Program - 8

```

import java.time.*;
class Demo18
{
    public static void main (String args[])
    {
        LocalDate ld = LocalDate.of(1997, 11, 11);
        System.out.println (ld.getMonth ());
        System.out.println (ld.getDayOfWeek ());
    }
}

```

Output :- November  
TUESDAY

while ( $c < ar.length$ ).  
}

{

$cr[c++] = ar[a++]$ ;

}

~~return cr;~~

}

~~program -10 //~~

class Demo

\$

public sum Cstring args[])

{

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

int res[] = merge (org);

for (int data : res)

{

s.o.println (data);

}

}

static int[] merge (int org[])

{

if (org.length == 1)

{

~~return org;~~

}

int cr[] = new int [org.length];

int ar[] = merge (Arrays.copyOfRange (org, 0, org.length / 2));

int br[] = merge (Arrays.copyOfRange (org, org.length / 2, org.length));

int a = 0; b = 0; c = 0;

```

while (a < ar.length && b < br.length)
{
    if (ar[a] < br[b])
        cr[c++] = ar[a++];
    else
    {
        cr[c++] = br[b++];
    }
}

while (b < br.length)
{
    cr[c++] = br[b++];
}

while (a < ar.length)
{
    cr[c++] = ar[a++];
}

return cr;
}

```

Output :-

1 2 3 4 5 6 8 9