

## Pattern Pointing

i)

\*\*\*\*\*

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

{

cout << "\*" << " ";

}

ii)

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

### Method - I

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

{

cout << "\*" << " ";

}

cout << endl;

// same loop 4 more times.

### Method - II

Nested  
loop  
(loop within  
another  
loop.)

{ for(i=1; i<=5; i++)  
{ for(j=1; j<=5; j++)  
{ cout << "\*" << " " ;  
}  
cout << endl;  
}

→ row 1

→ row 2

→ row 3

↓ ↓ ↓ ↓

Column 1

Col 2

Col 3

Col 4

row →  
 col | \* \* \* \* \*  
       \* \* \* \* \*  
       \* \* \* \* \*  
       \* \* \* \* \*  
       \* \* \* \* \*

- (i) row = 1
- (ii) row ≤ 5 ↙
- (iii) print \* 5 times
- (iv) r = row + 1

for (row = 1 ; row ≤ 5) {

    for (col = 1 ; col ≤ 5 ; col++)

        cout << \* ;

}

    cout << endl ;

}

iii)

10 10 10 10 10

10 10 10 10 10

10 10 10 10 10

10 10 10 10 10

(1) row = 1

(2) row = 4 ↙

(3) print 10 5 times

(4) row = row + 1

i) col = 1

ii) col ≤ 5 ↙

iii) print 10

iv) col = col + 1

code

for (row = 1 ; row ≤ 4 ; row = row + 1)

{

    for (col = 1 ; col ≤ 5 ; col++)

{

        cout << "10" << " " ;

}

    cout << endl ;

}

$\rightarrow$   
 $\overleftarrow{\text{row}}$

(IV)      col ↓ 1 1 1 1 1  
 2 2 2 2 2  
 3 3 3 3 3  
 4 4 4 4 4  
 5 5 5 5 5

(1) row = 1

(2) row <= 5

(3) print row 5 times

(4) row = row + 1

(a) col = 1

(b) col <= 5

(c) print row

(d) col = col + 1

### Code

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

{

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

{

```
        cout << "row" ; "
```

}

cout << endl ;

(V)      col ↓ 1 2 3 4 5  
 row → 1 2 3 4 5  
 1 2 3 4 5  
 1 2 3 4 5  
 1 2 3 4 5

(1) row = 1

(2) row <= 5

(3) print row to 5 times.

(4) row = row + 1

(a) col = 1

(b) col <= 5

(c) print col ;

(d) col = col + 1 ;

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

{

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

{

```
        cout << col ; "
```

cout << endl ;

}

Date \_\_\_\_\_  
Page \_\_\_\_\_

vi  $\xrightarrow{\text{col}}$

<u>row</u>	5	4	3	2	1
	5	4	3	2	1
	5	4	3	2	1
	5	4	3	2	1
	5	4	3	2	1

(1)  $\text{row} = 1$   
 (2)  $\text{row} <= 5$   
 (3) print 5 to 1 5 times  
 (4)  $\text{row} = \text{row} + 1$

a)  $\text{col} = 5$   
 b)  $\text{col} >= 1$   
 c) print col  
 d)  $\text{col}++$

for( $\text{row} = 1$ ;  $\text{row} <= 5$ ;  $\text{row}++$ )

{

for( $\text{col} = 5$ ;  $\text{col} >= 1$ ;  $\text{col}--$ )

{

$\text{cout} << \text{col} << " "$ ;

}

$\text{cout} << endl$ ;

}

vii  $\xrightarrow{\text{row}}$

$\text{col}$     1    4    9    16    25  
 1    4    9    16    25  
 1    4    9    16    25  
 1    4    9    16    25  
 1    4    9    16    25

H/W

viii

1    8    27    64    125  
 1    8    27    64    125  
 1    8    27    64    125  
 1    8    27    64    125  
 1    8    27    64    125

881

(ix) ~~rows~~ ~~cols~~

a	a	a	a	a
b	b	b	b	b
c	c	c	c	c
d	d	d	d	d
e	e	e	e	e

1 row = 1      char a' + (row - 1)

2 row <= 5

3 name = 'a' + (row - 1)

4 print name 5 times      for (row = 1; row <= 5; row++)  
                                   char name = 'a' + (row - 1);  
                                   for (col = 1; col <= 5; col++)  
                                   cout << name;

5 row++ ;

cout << endl;

3

(x) a b c d e      (xi) ~~rows~~

a	b	c	d	e
a	b	c	d	e
a	b	c	d	e
a	b	c	d	e
a	b	c	d	e

for (row = 1; row <= 5; row++)  
                                   for (col = 1; col <= 5; col++)  
                                   cout << col << " ";

for (col = 1; col <= 5; col++)  
                                   cout << endl;

3

(xi)

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

1 row = 1

2 row <= 5

3 cout = 1, print = count  
                                   count++

4 count++ count++

5 row++

Count : 1

```

for( row = 1 ; row <= 5 ; row = row + 1 )
    {
        for( col = 1 ; col <= 5 ; col++)
            {
                cout << count ;
                count++ ;
            }
        cout << endl ;
    }

```

Method - II

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

$$((row-1) \times 5) + 1$$

↑  
First element of every row.

$$((row-1) \times 5) + 5$$

↑  
Last element of every row.

$$((row-1) \times 5) + col$$

↑  
First element  
last element  
of every row

## Pattern Pointing - 2

(i)

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

① row = 1

② row &lt;= 5

③ Print '\*' row times

④ row = row + 1

① col = 1

② col &lt;= row

③ Print '\*'

④ col = col + 1

~~METHOD~~

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

```
    for(col=1; col<=row; col=col+1)
```

{

```
        cout << "*" << " " ;
```

{

```
    cout << endl;
```

{

~~METHOD~~

(ii)

1 2

1 2 3

1 2 3 4

1 2 3 4 5

① row = 1

② row &lt;= 5

③ Print 1 to row

④ row = row + 1

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

{

```
    for(col=1; col<=row; col++)
```

{

① col = 1

② col &lt;= row

③ print col

④ col = col + 1

{

```
    cout << col << " " ;
```

{

```
    cout << endl;
```

iii

1

2 2

i

$\text{row} = 1$

3 3 3

ii

$\text{row} \leq 5$

4 4 4 4

iii

Print  $\text{row}$ ,  $\text{row}$  times

5 5 5 5 5

iv

$\text{row} = \text{row} + 1$

for( $\text{row} = 1; \text{row} \leq 5; \text{row} = \text{row} + 1$ )  
{

i  $\text{col} = 1$

ii  $\text{col} < \text{row}$

for( $\text{col} = 1; \text{col} < \text{row}; \text{col} + +$ )  
{

iii  $\text{cout} \ll \text{row}$

iv  $\text{col} = \text{col} + 1$

$\text{cout} \ll \text{row} \ll \text{endl}$ ;

3

$\text{cout} \ll \text{endl};$

3

iv

1

2 1

i  $\text{row} = 1$

3 2 1

ii  $\text{row} \leq 5$

4 3 2 1

iii Print  $\text{row}$  to L

5 4 3 2 1

iv  $\text{row} = \text{row} + 1$

for( $\text{row} = 1; \text{row} \leq 5; \text{row} = \text{row} + 1$ )

1

for( $\text{col} = \text{row}; \text{col} >= 1; \text{col} = \text{col} - 1$ )

2

$\text{cout} \ll \text{col} \ll \text{endl}$ ;

3

$\text{cout} \ll \text{endl};$

3

(v)

a

b b

① row = 1

c c c

② row &lt;= 5

d d d d

③ print name row times

e e e e e

④ row = row + 1

row

1 → a

① col = 1

2 → b

② col &lt;= row

3 → c

③ name = 'a' + (row - 1)

⋮ ⋮

④ print name

for (row = 1; row &lt;= 5; row++)

⑤ col = col + 1;

{ char name = 'a' + (row - 1);

for (col = 1; col &lt;= row; col++)

2

char name = 'a' + (row - 1);

cout &lt;&lt; name &lt;&lt; " ";

3

cout &lt;&lt; endl;

3  
1 col

vi

row

1

2

3

4

5

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

\*

1

2

3

4

5

1

2

3

4

5

1

2

3

4

5

1

2

3

① row = 1

② row &lt;= 5

③ print '\*' (5 - (row + 1)) times

④ row = row + 1;

⑤ 5 - (row - 1)

or

5 - row + 1

for (row = 1; row &lt;= 5; row = row + 1)

{ for (col = 1; col &lt;= 5 - row + 1; col = col + 1)

{

cout &lt;&lt; "\*" &lt;&lt; " ";

3

cout &lt;&lt; endl;

{

vii) 1 2 3 4 5  
 1 2 3 4  
 1 2 3  
 1 2  
 1

- i) row = 1
- ii)  $x <= 5$
- iii) Print 1 to 5 - (row-1)
- iv)  $row = row + 1$

for (row = 1; row <= 5; row = row + 1)

{  
 for (col = 1; col <= 5 - (row - 1); col = col + 1)  
 {  
 cout << col << " "  
 }  
 cout << endl;  
}

viii) 5

5 4  
 5 4 3  
 5 4 3 2  
 1 → 1 → 5  
 2 → 2 → 5-4  
 3 → 3 → 5-3

- i) row = 1
- ii) row <= 5
- iii) print 5 to ~~5~~  
 $5 - (row - 1)$
- iv)  $row = row + 1$

for (row = 1; row <= 5; row = row + 1)

{  
 for (col = 5; col >= 5 - row + 1; col--)

{  
 cout << col << " "  
}

3  
 cout << endl;  
}

## Pattern Printing - 3

①

\* ~~n=5~~

	*	<del>n=5</del>	row	space	stars
	**		1	$5 - \text{row} = 4$	1
	***		2	$5 - \text{row} = 3$	2
	****		3	$5 - 3 = 2$	3
①	$\text{row} = 1$		4	$5 - 4 = 1$	4
②	$\text{row} <= 5$		5	$5 - 5 = 0$	5
③	Print space <del>n - row times</del>				
④	Print * row times				
⑤	$\text{row} = \text{row} + 1$				

for( $\text{row} = 1$ ;  $\text{row} <= n$ ;  $\text{row} = \text{row} + 1$ )

5

    for( $\text{col} = 1$ ;  $\text{col} <= 15 - \text{row}$ ;  $\text{col} = \text{col} + 1$ )

Σ

        cout << " ";

3

    for( $\text{col} = 1$ ;  $\text{col} <= \text{row}$ ;  $\text{col} = \text{col} + 1$ )

{

        cout << "\*";

3

    cout << endl;

3

(1)

1

2 2

Space → number

3 3 3

4 4 4 4	space $4 - \text{row} = 3$	row 1	num - print 1
i) row = 1	$4 - 2 = 2$	2	2
ii) row <= n	$4 - 3 = 1$	3	3
iii) print space $n - \text{row}$ times.	$4 - 4 = 0$	4	4
iv) print row, row times.			
v) row = row + 1			

for (row = 1; row <= n; row = row + 1)

{

    for (col = 1; col <= n - row; col = col + 1)

{

        cout << " ";

}

    for (col = 1; col <= row; col = col + 1)

{

        cout << row;

}

    cout << endl;

}

Q11

1  
1 2  
1 2 3       $n = \text{row} + \text{space}$   
1 2 3 4       $\text{Num} = \text{col} : (1 \rightarrow \text{row})$   
1 2 3 4 5

(i)  $\text{row} = 1$

(ii)  $\text{row} <= n$

(iii) Print space  $n - \text{row}$  times

(iv) Print 1 to  $\text{row}$

(v)  $\text{row} = \text{row} + 1$

$\text{for}(\text{row} = 1; \text{row} \leq n; \text{row} = \text{row} + 1)$

$\text{for}(\text{col} = 1; \text{col} \leq n - \text{row}; \text{col} = \text{col} + 1)$

$\text{cout} \ll " "$ ;

$\text{for}(\text{col} = 1; \text{col} \leq \text{row}; \text{col} = \text{col} + 1)$

$\text{cout} \ll \text{col} \ll " "$ ;

$\text{cout} \ll \text{endl};$

}

iv

A

A B

A B C

A B C D

A B C D E

Space → char

$(n - \text{row})$  At  $\text{A} + \text{row} - 1$

i)  $\text{row} = 1$

ii)  $\text{row} <= n$

iii) Print space  $n - \text{row}$  times

iv) Print 'A' to ' $\text{A}' + \text{row} - 1$

v)  $\text{row} = \text{row} + 1$

for ( $\text{row} = 1$ ;  $\text{row} <= n$ ;  $\text{row} = \text{row} + 1$ )

{

    for ( $\text{col} = 1$ ;  $\text{col} <= n - \text{row}$ ;  $\text{col} = \text{col} + 1$ )

{

        cout << " ";

}

    for ( $\text{col} = 'A'$ ;  $\text{col} = 'A' + \text{row} - 1$ ;  $\text{col} = \text{col} + 1$ )

{

        cout << col << " ";

}

    cout << endl;

}

(V)

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

Space  
(5-row)  
number  
row 401.

(I) row = 1

(II) row &lt;= n

(III) Print space narrow times

(IV) Print row to 1

(V) row = row + 1

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

```
    for (col=1; col<=n-row; col++)
```

{

```
        cout << " ";
```

3

```
    for (col=row; col>=1; col--)
```

{

```
        cout << *col << " ";
```

3

3

Pattern Printing Part-4Advance Pattern

①

\*

\*\* \*

\*\*\* \* \*

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

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

Space → *		
row	space	stars
1	4	1
2	3	3
3	2	5
4	1	7
0	0	9

i)  $\text{row} = 1$ ii)  $\text{row} \leq 5$ iii) point space  $5 - \text{row}$  timesiv) point "\*"  $2 \times \text{row} - 1$  timesv)  $\text{row} = \text{row} + 1$ 

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

{

```
    for (col=1; col<=n-row; col=col+1)
```

{

```
        cout << " ";
```

}

```
    for (col=1; col<=2*row-1; col=col+1)
```

{

```
        cout << "*" << " ";
```

}

}

(11)

1

n=5

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

Space → number.

(5-row)

 1 to row  
 increasing  
 (row-1) to 1  
 decreasing

① row=1

② row&lt;=5

③ print space 5-row times.

④ print 1 to row increasing order

⑤ print row-1 to 1 decreasing order.

⑥ row=row+1.

for (row=1; row&lt;=n; row=row+1)

for (col=1; col&lt;=n-row; col=col+1)

{

cout &lt;&lt; " ";

}

for (col=1; col&lt;=row; col=col+1)

cout &lt;&lt; col &lt;&lt; " ";

for (col=row-1; col&gt;=1; col=col-1)

cout &lt;&lt; col &lt;&lt; " ";

cout &lt;&lt; endl;

}

(iii)

\* \* \* \* \*

- \* \* \* \*

- - \* \* \*

- - - \* \*

n: 5

Space → \*

- - - - \*

row	space	*
5 1	0	9
4 2	1	7
3 3	2	5
2 4	3	3
1 5	4	1

$$\begin{aligned} & 2(n - \text{row}) + 1 \\ & 2(5 - 1) + 1 \\ & 8 + 1 = 9 \\ & 2(5 - 2) + 1 \\ & = 7 \\ & 2(n - \text{row}) + 1 \\ & 2(5 - 4) + 1 \\ & = 1 \end{aligned}$$

$$\begin{aligned} & 5 \times 2 - 2 \times 1 - 1 \\ & 5 \times 2 - 2 \times 1 \end{aligned}$$

$$\cancel{5 \times 2 - 2 \times 1}$$

$$\cancel{5 \times 2 - 2 \times 1}$$

↓  
row - 1  
Space

2 row - 1 → start  
 $\frac{9 - (2 \text{row} - 1)}{2}$  → space.

Method I

i)  $\text{row} = 5 \leftarrow n$

ii)  $\text{row} \geq 1$

iii) print space

$\frac{9 - (2 \text{row} - 1)}{2}$  times

iv) print \*  $\frac{2}{2 \text{row} - 1}$

v)  $\text{row} = \text{row} - 1$

$$\begin{aligned} & (2 \times n - 1) - 2(\text{row} - 1) \\ & = 2n - 2\text{row} = \frac{n - \text{row}}{2} \end{aligned}$$

Method - II

i)  $\text{row} = 1$

ii)  $\text{row} \leq n$

iii) print space

$\text{row} - 1$  times.

iv) print \*

$(2 \times (n - \text{row}) + 1)$  times.

v)  $\text{row} = \text{row} + 1$

Debug n=5

$(2 \times 5 - 1) + 1 \leftarrow \text{row} = 1 \times 5 \checkmark$   
 $\downarrow$   
 $(2 \times 5 - 2) + 1 \leftarrow \text{row} = 2 \times 5 \checkmark$   
 $\downarrow$   
 $(2 \times 5 - 3) + 1 \leftarrow \text{row} = 3 \times 5 \checkmark$   
 $\downarrow$   
 $(2 \times 5 - 4) + 1 \leftarrow \text{row} = 4 \times 5 \checkmark$

Method - 1

```
for (row = n; row >= 1; row = row - 1)
```

{

```
    for (col = 1; col <= n - row << col++)
```

{

```
        cout << " ";
```

}

```
    for (col = 1; col <= 2 * row - 1; col++)
```

{

```
        cout << "* ";
```

}

```
    cout << endl;
```

}

Method - 2

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

{

```
    for (col = 1; col <= row - 1; col++)
```

```
        cout << " ";
```

```
    for (col = 1; col <= (2 * (n - row) + 1); col++)
```

```
        cout << "* ";
```

```
    cout << endl;
```

}

IV

 $n=4$ 

Part - 1

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

$\star \rightarrow \text{SPACE} \rightarrow \star$

Part - 2

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

Part - 1

Space	row	Star	2xrow
0	4	4, 4	
2	3	3, 3	
4	2	2, 2	
6	1	1, 1	

Part - I

(I)  $row = 4, n$ (II)  $row \geq 1$ 

(III) Print \* row time

(IV) Print space  $2(n - row)$  time

(V) Print \* row time

(VI)  $row = row - 1$ 

Part - 2

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

print → space → print

Part - II

Space	row	Star	2xrow
6	1	1, 1	$\rightarrow 2 \times row$
4	2	2, 2	
2	3	3, 3	
0	4	4, 4	

 $0 - 2 \times row$  $2 \times n - 2 \times row$  $2(n - row)$

- i)  $\text{row} = 1$
- ii)  $\text{row} \leq n$
- iii) Print \*  $\text{row}$  times
- iv) Print space  $2(n - \text{row})$  times.
- v) Print \*  $\text{row}$  times.
- vi)  $\text{row} = \text{row} + 1$ .

`for (row=1; row >= 1; row--)`

`{ for (col=1; col <= row; col++)`

`cout << "*";`

`for (col=1; col <= 2(n-row); col++)`

`cout << " ";`

`for (col=1; col <= row; col++)`

`cout << "*";`

`cout << endl;`

3

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

S

`for (col=1; col <= row; col++)`

`cout << "*";`

`for (col=1; col <= 2(n-row); col++)`

`cout << " ";`

`for (col=1; col <= row; col++)`

`cout << "*";`

`cout << endl;`

3

(v)

part - 1

*		*
★ ★		★ ★
★ ★ ★		★ ★ ★
<hr/>		
★ ★ ★		★ ★ ★
★ ★		★ ★
*		*

$n = 4$

part - 2

★ → space → \*

part - 1

row	space	*	
1	6	1, 1	2 ← row
2	4	2 2	
3	2	3 3	
4	0	4 4	
	↓		

$2(n - \text{row})$

(i)  $\text{row} = 1$

(ii)  $\text{row} < n$

point \* row times

print space  $2(n - \text{row})$  times

(iii) print \* row times

(iv)  $\text{row} = \text{row} + 1$

part - 2

row	space	*	
3	2	3, 3	2 ← row
2	4	2, 2	
1	6	1, 1	

- 1  $\text{row} = n - 1$
- 2  $\text{row} \geq 1$
- 3 ~~-row print \* row times~~
- 4 ~~print space ( $2(n - \text{row})$ ) times~~
- 5 ~~Print \* row times~~
- 6  $\text{row} = \text{row} - 1$

$\text{for}(\text{row}=1; \text{row} \leq n; \text{row}++)$   
 {

$\text{for}(\text{col}=1; \text{col} \leq \text{row}; \text{col}++)$   
 $\text{cout} \ll \text{"*"};$

$\text{for}(\text{col}=1; \text{col} \leq 2 \times (n - \text{row}); \text{col}++)$   
 $\text{cout} \ll \text{" "};$

$\text{for}(\text{col}=1; \text{col} \leq \text{row}; \text{col}++)$   
 $\text{cout} \ll \text{"*"};$   
 $\text{cout} \ll \text{endl};$

3

$\text{for}(\text{row}=n-1; \text{row} \geq 1; \text{row}--)$

{

$\text{for}(\text{col}=1; \text{col} \leq \text{row}; \text{col}++)$   
 $\text{cout} \ll \text{"*"};$

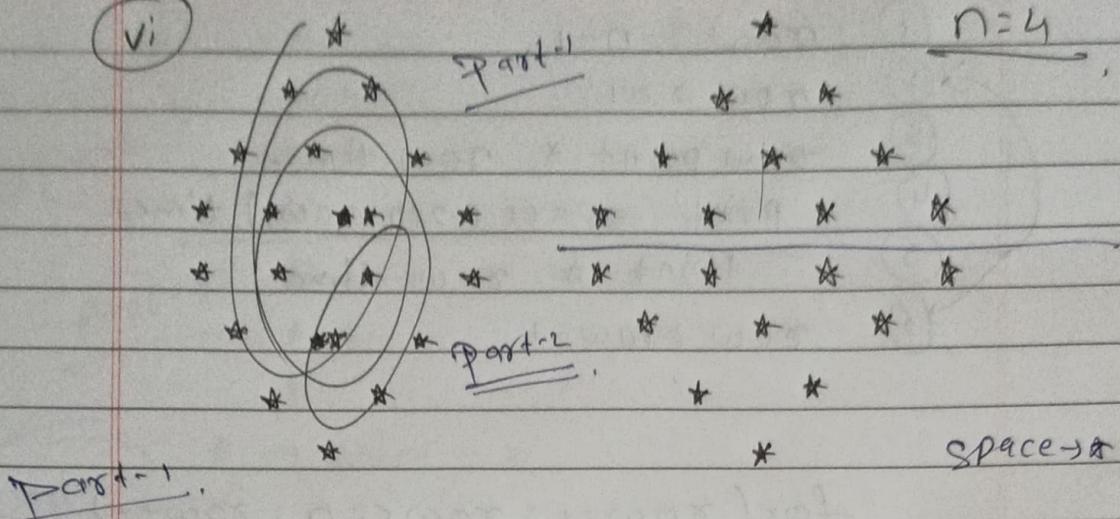
$\text{for}(\text{col}++; \text{col} \leq 2(n - \text{row}); \text{col}++)$   
 $\text{cout} \ll \text{" "};$

$\text{for}(\text{col}=1; \text{col} \leq \text{row}; \text{col}++)$   
 $\text{cout} \ll \text{"*"};$

$\text{cout} \ll \text{endl};$

3

vi



Part - 1.

row	star	space (-)
1	$1 \frac{(*)}{-}$	$4 - \text{row} = 3$
2	2	2
3	3	1
4	4	0

$n - \text{row}$

i)  $\text{row} = 1$

ii)  $\text{row} < n$

iii) print space  $n - \text{row}$  time

iv) print \* row times

v)  $\text{row} = \text{row} + 1$ .

Part - 2.

row	star	space
4	4	0
3	3	1
2	2	2
1	1	3

$n - \text{row}$

- ① row = n
- ② row >= 1
- ③ print - space n - row times.
- ④ print \* row times
- ⑤ row = row - 1.

```
for( row=1; row<=n; row++)  
{  
    for( col=1; col<=n-row; col=col+1)  
        cout << " ";  
    for( col=1; col<=row; col=col+1)  
        cout << "*";  
    cout << endl;  
}  
  
for( row=n; row>=1; row--)  
{  
    for( col=1; col<=n-row; col=col+1)  
        cout << " ";  
    for( col=1; col<=row; col=col+1)  
        cout << "*";  
    cout << endl;  
}
```

## Operators in C++

### BODMAS

Brackets Open Divide Multiplication  
Addition Subtraction

$$2 \times 3 - 4$$

$$6 - 4 = 2$$

$$2 \times 3 - 4 / 2$$

$$6 - 2 = 4$$

### Arithmetic operators

Unary

Binary

+

-

x

% (Rem) modulus.

Operator

$$2 + 3 = 5$$

Operand      Operand

$$\frac{\text{float}}{\text{int}} = \text{float}$$

$$\frac{\text{int}}{\text{int}} = \text{int}$$

$$16/4 = 4$$

$$18/5 = 3$$

$$18/4 = 4$$

$$\frac{12 \cdot 4}{4} = \underline{\underline{3 \cdot 1}}$$

$$63/4 = 0.3$$

$$\begin{array}{r} 12.6 \\ \times 3 \\ \hline 43.8 \end{array}$$

{ \* / ÷ } > { +, - }

Left to Right

Precedence

Associativity

Precedence

int < float < double

e.g. ①  $3 + 4 - 6 = 2$

$$7 - 6 = 2$$

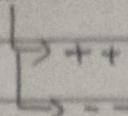
$$1 - 2 = \textcircled{-1}$$

②  $2 \times 3 - 4 / 2$

$$6 - 2$$

$$\textcircled{4}$$

### unary operator



①  $++$

a) post increment :  $a++ \rightarrow a = a+1$

b) pre increment :  $++a \rightarrow a = a+1$

e.g. ①  $\begin{matrix} a=10 \\ b = \cancel{a++} \end{matrix}$

$$b = 10$$

$$a = 11$$

②  $\begin{matrix} a=10 \\ b = ++a \end{matrix}$

$$a = 11$$

$$b = 11$$

first assignment,  
then increment.

first increment,  
then assignment

②  $--$

a) post decrement :  $a-- \rightarrow a = a-1$

b) pre decrement :  $--a \rightarrow a = a-1$

e.g. ①  $\begin{matrix} a=10 \\ b = a-- \end{matrix}$

$$b = 10$$

$$b = 9$$

first assignment  
then decrement

②  $\begin{matrix} a=10 \\ b = --a \end{matrix}$

$$a = 9$$

$$b = 9$$

First ~~decrement~~  
then assignment

## Comparison Operators

$\{ ==, >, <, \geq, \leq, != \}$

Answer of comparison operator is

yes or no i.e. 1 or 0 (boolean).

(i)  $= =$

e.g.

$$\textcircled{(i)} \quad 3 == 4 = 0 \quad \textcircled{(ii)} \quad 4 == 4 = \textcircled{1} \quad \begin{matrix} \uparrow \\ \text{false} \end{matrix} \quad \begin{matrix} \uparrow \\ \text{true} \end{matrix}$$

(ii)  $>$

$$\text{e.g. } \textcircled{(i)} \ 10 > 5 = 1, \quad \textcircled{(ii)} \ 10 > 20 = 0$$

(iii)  $<$

$$\text{e.g. } \textcircled{(i)} \ 10 < 5 = 1$$

$$\textcircled{(ii)} \quad \begin{matrix} s > 4 > 3 \\ \overbrace{1 > 3} \\ \textcircled{0} \leftarrow \end{matrix} \quad \text{left to right.}$$

(iv)  ~~$\neq$~~   $\neq$

$$\text{e.g. } \textcircled{(i)} \ 4 \neq 4 = 1, \quad \textcircled{(ii)} \ 8 \neq 4 = 1$$

$$\textcircled{(iii)} \ 10 <= 8 = 0$$

(v)  $!=$

$$\text{e.g. } \textcircled{(i)} \ 4 != 5 = 1, \quad \textcircled{(ii)} \ 5 != 5 = 0$$

(vi)  $<=$

$$\text{e.g. } \textcircled{(i)} \ 4 <= 5 = 1$$

$$\textcircled{(ii)} \ 8 <= 7 = 0$$

Precedence

$$\{ >, <, \geq, \leq \} > \{ ==, != \}$$

left to right  $\leftarrow$  associativity.

e.g.

$$\textcircled{1} \quad 5 > 4 < 3 == 2$$

$$1 < 3 == 2$$

$$1 == 2$$

$$\textcircled{0} \quad A$$

$$\textcircled{1} \quad 3 > 4 > 5 !=$$

$$0 > 5$$

$$0 != 1$$

$$\textcircled{1} \quad B$$

### Logical operators

(  $\&$ ,  $\|$ ,  $!$  )

AND OR

Not

A B AND( $\&$ )

e.g.:-

2	4	4	3	,	0 & 4	0	0	0
				,	0 & 4	0	1	0
				,	0 & 4	1	0	0

~~0 & 1 = 1~~

~~0 || 5 = 1~~

Non-zero is considered as true.

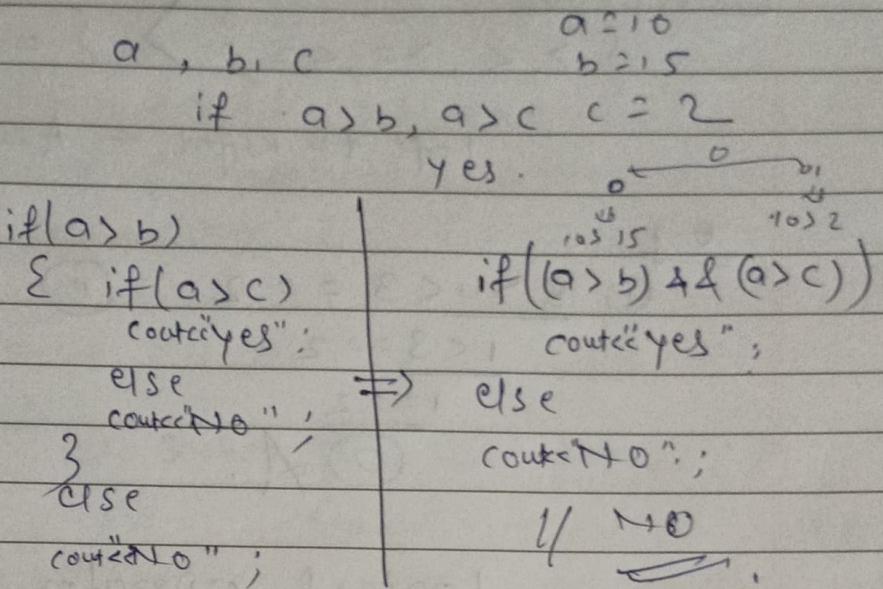
A	B	OR
0	0	0
0	1	1
1	0	1

$!S = 0$

$!0 = 1$

A	NOT
0	1
1	0

e.g :-



e.g

vowel or consonant.

char name = 'a';

```

if (name == 'a' || name == 'e' || name == 'i'
    || name == 'o' ; name == 'u')
  
```

Σ

cout << "vowel";

3

else

cout << "consonant";

### Ø SHORT Circuit AND(&&)

The expression is evaluated until we get one false result, because if we do AND(&&) operation with '0' result will be '0' always, no further expression will be evaluated.

## SHORT circuit (OR)

The expression is evaluated until we get first one 'true'(1) result because we do OR(11) with true(1) result will will have always, so no further expression will be evaluated.

## Bitwise Operators

& , | , ^ , ~ , << , >> }  
 Bitwise And      Bitwise OR      XOR      Complement      Left shift      Right shift

e.g. ①  $2 \& 3 = (2)$       ②  $2 | 3 = (3)$

$$\begin{array}{r}
 \downarrow \\
 \begin{array}{r}
 10 \\
 \& 11 \\
 \hline
 10 \leftarrow 2
 \end{array}
 \qquad \qquad \qquad
 \begin{array}{r}
 10 \\
 111 \\
 \hline
 11 \leftarrow 3
 \end{array}
 \end{array}$$

③  $\wedge$  (XOR)

A	B	$\wedge$ (XOR)	$2 \wedge 3 = (1)$
0	0	0	
0	1	1	$10$
1	0	1	$11$
1	1	0	$01 \leftarrow 1$

(iv)

&lt;&lt; (left shift)

$$6 << 1 = 12 \rightarrow 6 \times 2^1 = 12.$$

↓ Left Shift, one time.

0 ... 000110

$\swarrow \searrow \swarrow \searrow$

1 100 < 12

6 << 2 = 24 if we do left shift, on every shift value get double.

$$\dots 0001100 \rightarrow 6$$

$\swarrow \searrow \swarrow$

$$0 \dots 0001100 \rightarrow 12$$

 $(2^n)$ 

(num  $\times 2^n$ )

$$6 << 2$$

$\swarrow \searrow$

$$6 \times 2^2 = 6 \times 4 = 24.$$

- 6 << 2

X

&lt;&lt;, only for +ve.

(v)

&gt;&gt;, right shift

6 &gt;&gt; 1

↓

$\swarrow \searrow$

011 &lt; 3

num  $\times 2^n$

if we do right shift, on every shift value get half.

VI

$\sim$  (complement)

$$\begin{array}{ccc} \sim & & \sim \\ \sim 5 & & (0 \rightarrow 1) \\ 00...0000101 & & 1 \rightarrow 0 \\ \xrightarrow{\text{Ans.}} 11\ 1111010 & & \end{array}$$

$$\begin{array}{r} 1's \ 0000101 \\ + 1 \\ \hline 00000110 \end{array}$$

(-6)

$$\sim 5 \rightarrow -6$$

$$\sim 8 = -9$$

$$\sim 13 \rightarrow -14$$

~~Precedence~~

$\{ \ll, \gg \} > \{ \cdot, , , ^ \}$

(Assignment operators)

$\{ +=, -=, *=, /=, \%=\}$

$$a *= 3 \leftarrow a = a * 3$$

$$a /= 2 \leftarrow a = a / 2$$

Refer : Precedence table from gfg.