

Patterns

(Class Slides)

Note: We are sharing these slides as lot of mathematical logic building was done on paper.

Hope these will help you in your learning process ↗

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

A 4x4 grid of yellow asterisks on a dark background.

A grid of 10 yellow asterisks arranged in three rows: top row has 4 stars, middle row has 3 stars, bottom row has 2 stars.

1
12
123
1234

A
B C
D E F
G H I J

A grid of 15 yellow five-pointed star icons arranged in four rows. The first row has five stars. The second row has three stars. The third row has two stars. The fourth row has four stars.

A triangular arrangement of six yellow asterisks, forming three rows of two asterisks each.

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

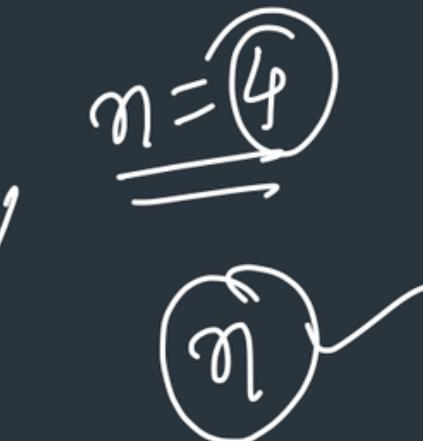
The image features a dark gray background with a central decorative element. This element consists of a grid of yellow asterisks (*). The grid is organized into seven horizontal rows. The first row contains a single asterisk at the top center. The second row has three asterisks. The third row has five asterisks. The fourth row has seven asterisks. The fifth row has five asterisks. The sixth row has three asterisks. The seventh and final row contains a single asterisk at the bottom center. The asterisks are a bright yellow color and are evenly spaced within each row.

A 7x7 grid of yellow asterisks ('*') on a dark blue background. The asterisks are arranged in a pattern where they form a central column and a central row, with some missing from the corners and the center.

Nested Loops

Loop inside a loop

	C1	C2	C3	C4
Row1	1	1	1	1
R2	2	2	2	2
R3	3	3	3	3
R4	4	4	4	4



① Outer loop : no. of Rows

② Inner loop : no. of columns /
each row

③ Work in inner loop (each row)

i point

```
for(int i=1; i<=4; i++) {  
    for(int j=1; j<=4; j++) {  
        cout << i << " " << j  
    }  
}
```

Print Star pattern

R1	*	1st
R2	**	2 st
R3	***	3 st
R4	****	4 st

$$\underline{\underline{n=4}}$$

① outer loop → Rows (i) → Row no.
n times (1 to n)

② inner loop (each rows)
columns
i times (1 to i)

③ work?
cout << ""

```
for (int i=1 ; i<=n ; i++) {  
    for (int j=1 ; j<=i ; j++) {  
        cout << "*"  
    }  
    cout << endl;  
}  
n = 4
```

Print Inverted Star pattern

R₁ * * * *
R₂ * * *
R₃ * *
R₄ *

$$\begin{aligned}n &= 4 \\4 &= n - i + 1 \\3 &= n - i + 1 \\2 &= n - i + 1 \\1 &= n - i + 1\end{aligned}$$

n = 4

① outer loop (rows)
(1 to n)

② inner loop (each row)
(1 to n-i+1)

③ work?
cout << "*"

Print Half Pyramid pattern

R1	1
R2	12
R3	123
R4	1234

n = 4

1 to i

```
for(int i=1; i<=n; i++) {  
    for(int j=1; j<=i; j++) {  
        cout << j ;  
    }  
    end line  
}
```

n = 4

① outer loop (rows)

(1 to n)

② inner loop (each row)

(1 to i) $\Rightarrow j$

③ work?

cout << j ;

Print Character Pyramid pattern

~~A
BC
DEF
GHIJ~~

$n = 4$
char $ch = 'A'$

$n = 4$

① outer loop (row)
(1 to n)

② inner loop (each row)
(1 to i)

③ work?

`cout << ch ;
ch ++ ; //`

Print Hollow Rectangle pattern

R1		1st + 3st + 1st
R2		1st + 3sp + 1st
R3		1st + 3sp + 1st
R4		1st + 3st + 1st

first or last \rightarrow 5 stars
(1) (n)

n = 4

① Outer loop (rows)
(1 to n)

② Inner loop (each row)

```
cout << "*" ; //First
for ( 1 to n-1 ) {
    1st or last  $\rightarrow$  "*"
    else  $\rightarrow$  " "
}
cout << "*" ; //last
```

Inverted & Rotated Half-Pyramid

n=4

— — — *

— — * *

- * * *

* * * *

R1 3 sp + 1st
R2 2 sp + 2nd
R3 1 sp + 3rd
R4 0 sp + 4th

$n - i$ spaces

i=1

$4 - 1 = 3$

i=2

$4 - 2 = 2$

i=3

$4 - 3 = 1$

i=4

$4 - 4 = 0$

n = 4

① outer loop (rows)
(1 to n)

② inner loop (each row)

a) Spaces (1 to $n-i$)
cout << " " work

b) Stars (1 to i)
cout << "*" work

cout << endl

Print Floyd's Triangle

R1	1	1 el
R2	2 3	2 el
R3	4 5 6	3 els
R4	7 8 9 10	4 el
R5	11 12 13 14 15	5 el

*i*th i times
 $(1 \text{ to } i)$

num=1

① outer loop (rows)

(1 to n)

② inner loop (each row elements)

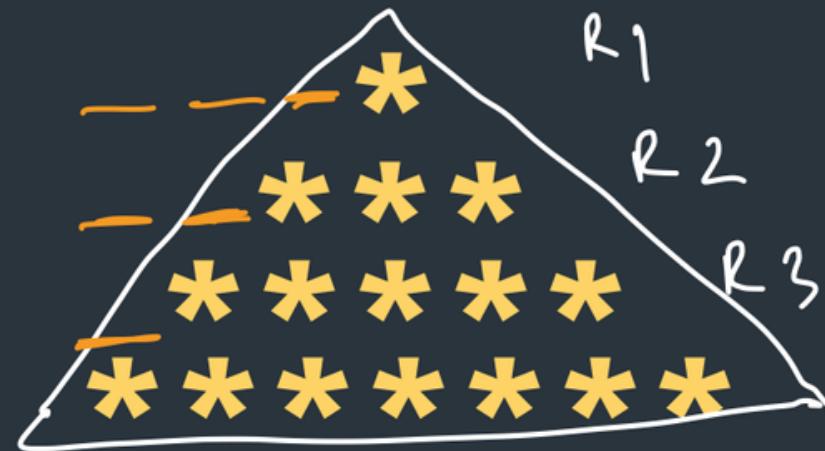
(1 to i)

③ work?

```
cout << num;
num++;
```

Diamond Pattern

$n=4$



$3sp + 1st$
 $2sp + 3st$
 $1sp + 5st$
 $0sp + 7st$

1st pyramid
 ① outer loop (rows)
 (1 to n)

② inner loop (each row)
 elements

a) (1 to $n-i$) cout << "

b) (1 to $2*i-1$) cout << "*"

$n = 4$

$$i=1 \rightarrow 1$$

$$i=2 \rightarrow 3$$

$$i=3 \rightarrow 5$$

$$i=4 \rightarrow 7$$

$$\frac{2*i-1}{2} = 1$$

$$\frac{2*i-1}{2} = 3$$

$$\frac{2*i-1}{2} = 5$$

$$\frac{2*i-1}{2} = 7$$

Diamond Pattern

$n=4$

**** * * * R₄
-* * * * * R₃
-- * * * R₂
--- * R₁

$n = 4$

0 sp + 7 st
1 sp + 5 st
2 sp + 3 st
3 sp + 1 st

$2 \times i - 1$

- 2nd pyramid
- ① outer loop (rows)
 $(n \text{ to } 1)$
 - ② inner loops (each row elements)
 - a) sp ($i \text{ to } n-i$)
 - b) st ($1 \text{ to } 2 \times i - 1$)

Print Butterfly Pattern

$$\begin{array}{ccccc}
 1st & + & 6sp & + & 1st \\
 2st & + & 4sp & + & 2st \\
 3st & + & 2sp & + & 3st \\
 4st & + & 0sp & + & 4st \\
 \downarrow & & \downarrow & & \downarrow \\
 2*(n - i)
 \end{array}$$

A 6x6 grid of asterisks ('*') arranged in a diamond pattern. The asterisks are yellow on a dark blue background. The pattern consists of six rows: the first and last rows have two asterisks; the second and fifth rows have four asterisks; the third and fourth rows have five asterisks; and the center row has six asterisks.

Pattern 1

① outer loop (rows)

(1 to n)

② inner loop

a) starts (1 to i)

b) spaces ($1 \text{ to } 2^{*(n-i)}$)

c) Stars (1 to i)

$$n = 4$$

Print Butterfly Pattern

$n=4$

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

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

Pattern

① outer loop (rows) (n to 1)

② inner loop

a) stars (1 to i)

b) spaces (1 to $2*(n-i)$)

c) stars (1 to i)

R4 4 st ✓
R3 3 st ✓
R2 2 st ✓
R1 1 st ✓

$$2 * (n - i) = 2 * (4 - 1) = 6$$

i = 1

$n = 4$