

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 dark blue square containing a 4x4 grid of yellow asterisks (*). The asterisks are arranged in four horizontal rows, with each row containing four asterisks.

A dark blue rectangular background featuring a grid of twelve yellow asterisks. The asterisks are arranged in four horizontal rows, with each row containing three asterisks. They are positioned in the upper half of the image.

1
12
123
1234

A
B C
D E F
G H I J

A grid of 15 yellow five-pointed stars arranged in three rows of five. The stars are evenly spaced and have a bright yellow color.

A 4x4 grid of yellow asterisks arranged in four rows and four columns, centered on the page.

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

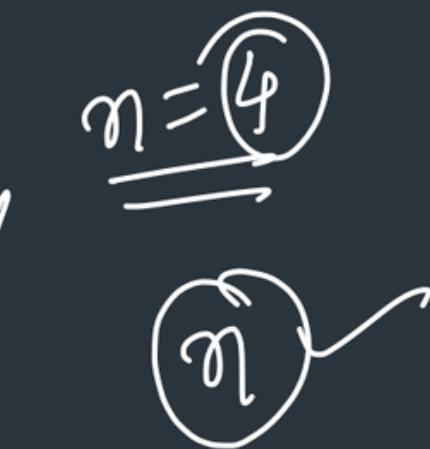
A 7x7 grid of yellow asterisks arranged in seven rows and seven columns, centered on a black background.

A 6x6 grid of yellow asterisks on a dark background. The asterisks are arranged in six rows and six columns. The first row has one asterisk at the top-left and one at the top-right. The second row has two asterisks in the middle-left and two in the middle-right. The third row has three asterisks in the middle-left. The fourth row has four asterisks in the middle-left. The fifth row has five asterisks in the middle-left. The sixth row has six asterisks in the middle-left.

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	1 2
R3	1 2 3
R4	1 2 3 4

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 "*" } ②
else \rightarrow " "

} cout << "*" ; //last

Inverted & Rotated Half-Pyramid

$n=4$

---	*	R1 3 sp + 1st
--	**	R2 2 sp + 2st
-	***	R3 1 sp + 3st
****		R4 0 sp + 4st

$i = 1 \quad n - i = 3$
 $i = 2 \quad n - 2 = 2$
 $i = 3 \quad n - 3 = 1$
 $i = 4 \quad n - 4 = 0$

$n = 4$

- 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 el		
R4	7	8	9	10	4 el	
R5	11	12	13	14	15	5 el

i^{th} i times
 (1 to i)

num=1

① outer loop (rows)

(1 to n)

② inner loop (each row elements)

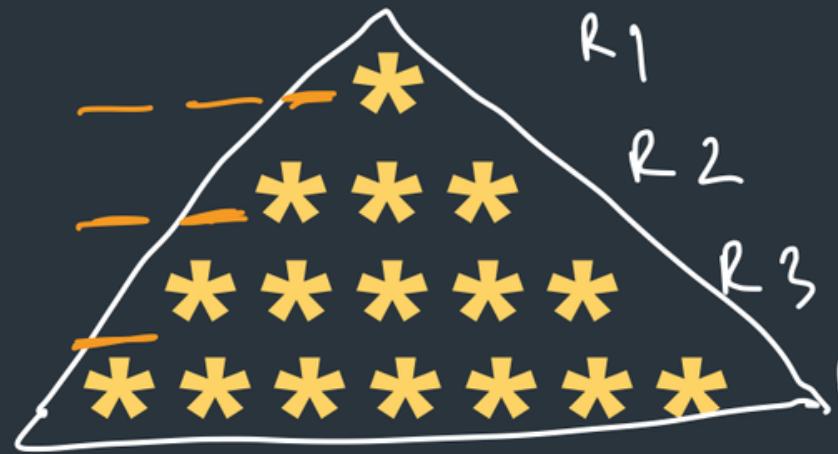
(1 to i)

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 << "*"

$n = 4$

$$\begin{array}{ll}
 i=1 \rightarrow 1 & \underline{2 \times 1 - 1} = 1 \\
 i=2 \rightarrow 3 & \underline{2 \times 2 - 1} = 3 \\
 i=3 \rightarrow 5 & \underline{2 \times 3 - 1} = 5 \\
 i=4 \rightarrow 7 & \underline{2 \times 4 - 1} = 7
 \end{array}$$

Diamond Pattern

$n=4$

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

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

2*i - 1

2nd pyramid
① outer loop (rows)
($n \rightarrow 1$)

② inner loops (each row elements)
a) sp ($1 \text{ to } n-i$)

b) st ($1 \text{ to } 2*i-1$)

$n = 4$

Print Butterfly Pattern



$n=4$



$$\begin{aligned}
 R1 &: 1st + 6sp + 1st \\
 R2 &: 2st + 4sp + 2st \\
 R3 &: 3st + 2sp + 3st \\
 R4 &: 4st + 0sp + 4st
 \end{aligned}$$



$$2 \times (n - i)$$

$n = 4$

Pattern

① outer loop (rows)
(1 to n)

② inner loop

a) stars (1 to i)

b) spaces (1 to $2 \times (n - i)$)

c) stars (1 to i)

Print Butterfly Pattern

$n=4$

```

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

```

```

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

```

Pattern

① outer loop (rows) ($n \text{ to } 1$)

② inner loop

a) stars (1 to i)

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

c) stars (1 to i)

✓
 0 sp 4 st
 2 sp ✓ 3 st
 4 sp ✓ 2 st
 6 sp ✓ 1 st

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

$n = 4$

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

$i = 1$