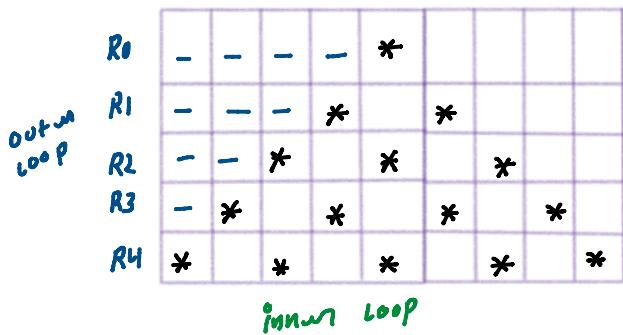


class : 4th — 31 / 08 / 23
<https://www.linkedin.com/in/manojofficialmj/>

PATTERN PROBLEMS

- Step 01: Count the numbers of row (outer loop)
 Step 02: See what is happening in each row (inner loop)

PATTERN:08 Full pyramid pattern



CODE OF PATTERN 08

```

1 // Pattern 08 --> Full Pyramid Pattern
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     cin>>n;
8
9     // outer loop
10    for(int row=0; row<n; row++){
11        // print space
12        for(int col=0; col<n-row-1; col++){
13            cout << " ";
14        }
15        // print star
16        for(int col=0; col<row+1; col++){
17            cout << "* ";
18        }
19        // new Line
20        cout<<endl;
21    }
22
23    return 0;
24 }
```

Step 01: Count the numbers of row (outer loop)
 Numbers of row = $n = 5$ ($row < h$)

Step 02: See what is happening in each row (inner loop)

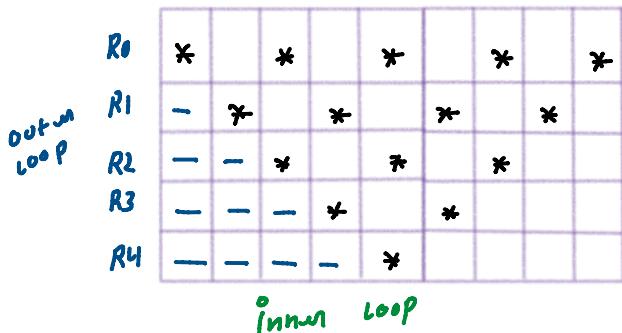
(Row + 1) $\rightarrow (h - row - 1)$

Row0 = print 4space 1star	$= 5 \text{ char}$
Row1 = print 3space 2star	$= 5 \text{ char}$
Row2 = print 2space 3star	$= 5 \text{ char}$
Row3 = print 1space 4star	$= 5 \text{ char}$
Row4 = print 0space 5star	$= 5 \text{ char}$

PATTERN:09 Inverted Full pyramid pattern



Step 01: Count the numbers of row (outer loop)



CODE OF PATTERN 09

```

1 // Pattern 09: Inverted Full Pyramid
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     cin>>n;
8
9     // outer loop
10    for(int row=0; row<n; row++){
11        // print space
12        for(int col=0; col<row; col++){
13            cout << " ";
14        }
15        // print star
16        for(int col=0; col<n-row; col++){
17            cout << "* ";
18        }
19        // new Line
20        cout<<endl;
21    }
22    return 0;
23 }
```

Step 01: Count the numbers of row (outer loop)

Numbers of row = $n = 5$ ($\text{row} < h$)

Step 02: See what is happening in each row (inner loop)

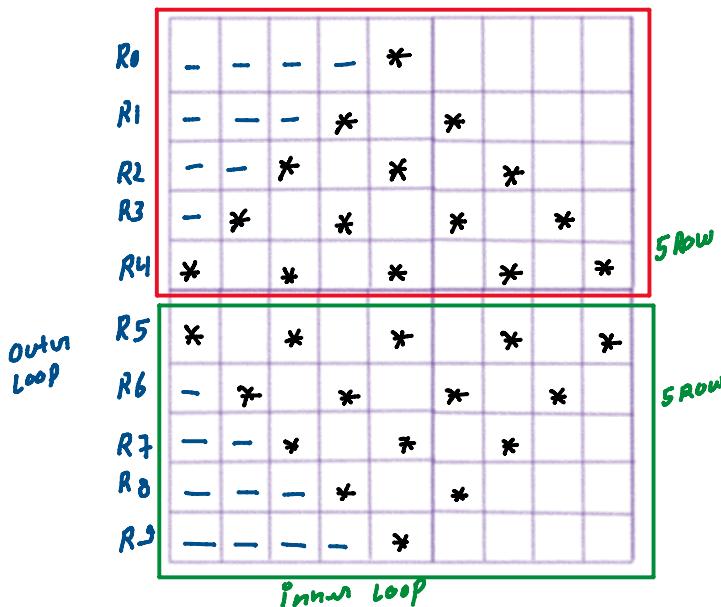
$(n - \text{row})$

Row0 = print 0space 5star
Row1 = print 1space 4star
Row2 = print 2space 3star
Row3 = print 3space 2star
Row4 = print 4space 1star

$= 5 \text{ char}$
 $= 5 \text{ char}$
 $= 5 \text{ char}$
 $= 5 \text{ char}$
 $= 5 \text{ char}$

(row)

PATTERN:10 Dimond pattern



Step 01: Count the numbers of row (outer loop)

Numbers of row = $n = 10$ ($\text{row} < h$)

Step 02: See what is happening in each row (inner loop)

$h = 10$

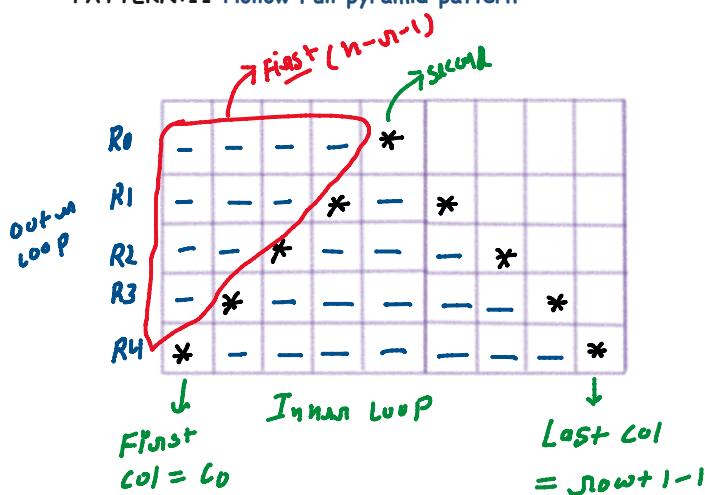
1. First print full pyramid patter
2. Second print inverted full pyramid pattern

CODE OF PATTERN 10

```

1 // Pattern 10 --> Dimond Pattern
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int num;
7     cin>>num;
8     int n = num/2;
9
10    //TODO 01: Full Pyramid
11    for(int row=0; row<n; row++){
12        // print space
13        for(int col=0; col<n-row-1; col++){
14            cout << " ";
15        }
16        // print star
17        for(int col=0; col<row+1; col++){
18            cout << "* ";
19        }
20        // new line
21        cout<<endl;
22    }
23    //TODO 02: Inverted Full Pyramid
24    for(int row=0; row<n; row++){
25        // print space
26        for(int col=0; col<row; col++){
27            cout << " ";
28        }
29        // print star
30        for(int col=0; col<n-row; col++){
31            cout << "* ";
32        }
33        // new line
34        cout<<endl;
35    }
36
37    return 0;
38 }
```

PATTERN:11 Hollow Full pyramid pattern



CODE OF PATTERN 11

```

1 // Pattern 11 --> Hollow Full Pyramid
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     cin>>n;
8
9     // outer loop
10    for(int row=0; row<n; row++){
11        // TODO 01: First
12        for(int col=0; col<n-row-1; col++){
13            cout << " ";
14        }
15        // TODO 02: Second
16        for(int col=0; col<row+1; col++){
17            if(col==0 || col==row+1-1){
18                cout << "** ";
19            }
20            else{
21                cout << " ";
22            }
23        }
24        cout << endl;
25    }
26 }
```

Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 5 ($\text{row} < h$)

Step 02: See what is happening in each row (inner loop)

First $(n - \text{row} - 1)$

Row - us

R0 - 4S
R1 - 3S
R2 - 2S
R3 - 1S
R4 - 0S

Second $\star | \text{col} == 0 \text{ || col } == \text{row} + 1 - 1$

$R0 - 1*$	$= 1ch$
$R1 - 2* 1S 1*$	$= 3ch$
$R2 - 1* 3S 1*$	$= 5ch$
$R3 - 1* 5S 1*$	$= 7ch$
$R4 - 1* 7S 1*$	$= 9ch$

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

```

1 // Pattern 11 --> Hollow Full Pyramid
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     cin>>n;
8
9     // outer loop
10    for(int row=0; row<n; row++){
11        // TODO 01: First
12        for(int col=0; col<n-row-1; col++){
13            cout << " ";
14        }
15        // TODO 02: Second
16        for(int col=0; col<row+1; col++){
17            if(col==0 || col==row+1-1){
18                cout << "* ";
19            }
20            else{
21                cout << " ";
22            }
23        }
24        // new Line
25        cout<<endl;
26    }
27    return 0;
28 }

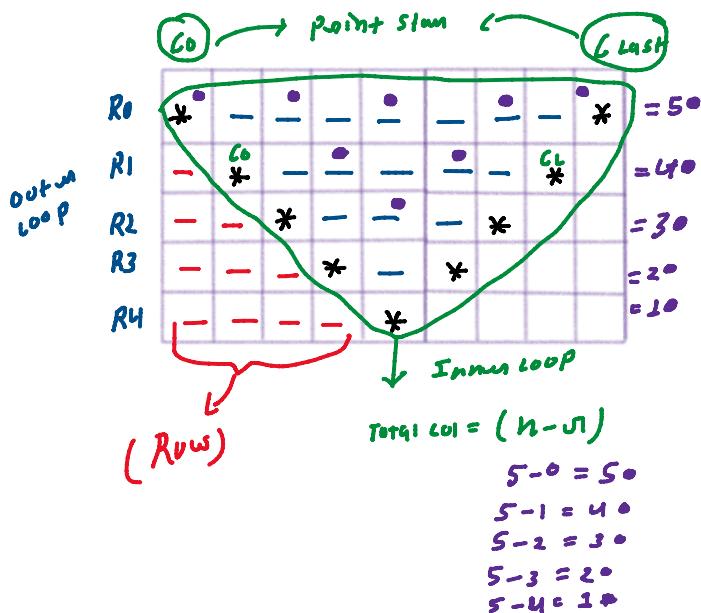
```

→ Second Star | $col == 0 \text{ or } col == n - 1$

$R_0 - 1*$	$= 1ch$
$R_1 - 2* 2S 2*$	$= 3ch$
$R_2 - 1* 3S 1*$	$= 5ch$
$R_3 - 1* 5S 1*$	$= 7ch$
$R_4 - 1* 7S 1*$	$= 9ch$

$\} (n + 1)$

PATTERN:12 Inverted Hollow Full pyramid pattern



Step 01: Count the numbers of row (outer loop)

Numbers of row = $n = 5$ ($row < h$)

Step 02: See what is happening in each row (inner loop)

→ First (row)

$R_0 - 0S$
$R_1 - 1S$
$R_2 - 2S$
$R_3 - 3S$
$R_4 - 4S$

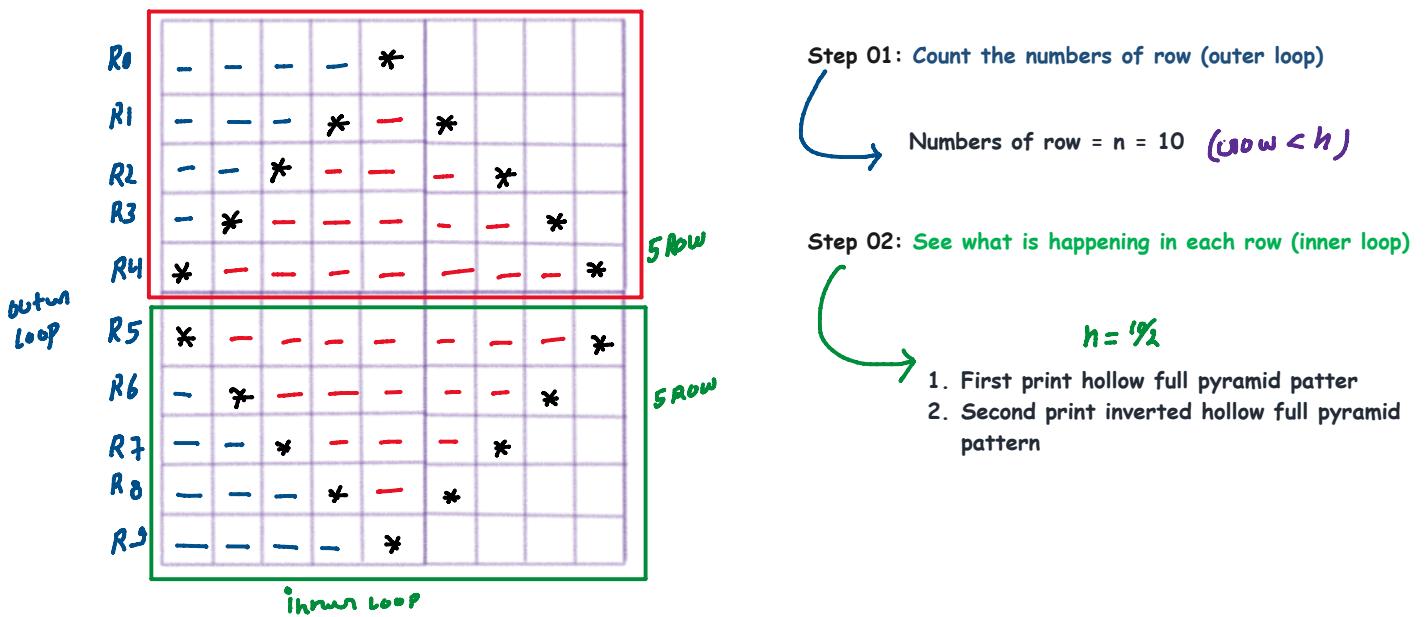
CODE OF PATTERN 12

```

1 // Pattern 12 --> Inverted Hollow Full Pyramid
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6     int n;
7     cin>>n;
8
9     // outer loop
10    for(int row=0; row<n; row++){
11        // TODO 01: First
12        for(int col=0; col<row; col++){
13            cout << " ";
14        }
15        // TODO 02: Second
16        for(int col=0; col<n-row; col++){
17            if(col==0 || col==(n-row)-1){
18                cout << "* ";
19            }
20            else{
21                cout << " ";
22            }
23        }
24        // new Line
25        cout<<endl;
26    }
27    return 0;
28 }

```

PATTERN:13 Hollow Dimond pattern



CODE OF PATTERN 13

```

// Pattern 13 --> Hollow Diamond pattern
#include<iostream>
using namespace std;

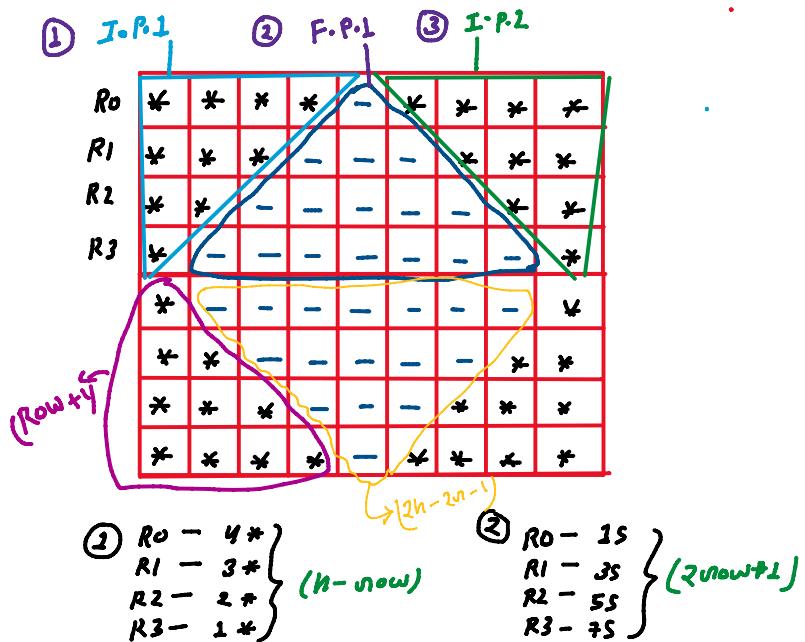
int main(){
    int num;
    cin>num;
    int n = num/2;

    // * Hollow Full Pyramid
    for(int row=0; row<n; row++){
        // TODO 01: First
        for(int col=0; col<n-row-1; col++){
            cout << " ";
        }
        // TODO 02: Second
        for(int col=0; col<row+1; col++){
            if(col==0 || col==row+1-1){
                cout << "* ";
            }
            else{
                cout << " ";
            }
        }
        // new line
        cout<<endl;
    }
    // * Inverted Hollow Full Pyramid
    for(int row=0; row<n; row++){
        // TODO 01: First
        for(int col=0; col<row; col++){
            cout << " ";
        }
        // TODO 02: Second
        for(int col=0; col<n-row; col++){
            if(col==0 || col==(n-row)-1){
                cout << "* ";
            }
            else{
                cout << " ";
            }
        }
        // new line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n = 10
      *
     * *
    *   *
   *     *
  *       *
 *         *
*           *
*             *
*               *
*                 *
*                   *
*                     *
*                       *
*/

```

PATTERN:14 Flipped Solid Diamond pattern



Step 01: Count the numbers of row (outer loop)

Numbers of row = $n = 4$ ($\text{row} < h$)

Step 02: See what is happening in each row (inner loop)

1. First print inverted pyramid patter
2. Second print full pyramid pattern
3. Third print inverted pyramid patter

$$\begin{cases} R_0 = 4* \\ R_1 = 3* \\ R_2 = 2* \\ R_3 = 1* \end{cases} \quad (n - \text{row})$$

CODE OF PATTERN 14

```

// Pattern 14 --> Flipped Solid Diamond pattern
#include<iostream>
using namespace std;

int main(){
    int num;
    cin>>num;
    int n = num/2;

    //? FIRST PART
    for(int row=0; row<n; row++){
        // TODO 01: inverted pyramid 1
        for(int col=0; col<n-row; col++){
            cout << "* ";
        }
        // TODO 02: full pyramid
        for(int col=0; col<2*row+1; col++){
            cout << "- ";
        }
        // TODO 03: inverted pyramid 2
        for(int col=0; col<n-row; col++){
            cout << "* ";
        }
        // new line
        cout<<endl;
    }
    //? SECOND PART
    for(int row=0; row<n; row++){
        // TODO 01: inverted pyramid 1
        for(int col=0; col<row+1; col++){
            cout << "* ";
        }
        // TODO 02: full pyramid
        for(int col=0; col<2*n-2*row-1; col++){
            cout << "- ";
        }
        // TODO 03: inverted pyramid 2
        for(int col=0; col<row+1; col++){
            cout << "* ";
        }
        // new line
        cout<<endl;
    }
    return 0;
}

/*
OUTPUT: when n = 8
* * * * - * * *
* * * - - - * * *
* * - - - - - * *
* - - - - - - - *
* * - - - - - - * *
* * * - - - - * *
* * * * - - * * *
* * * * * - * * *
*/

```

PATTERN:15 Fancy pattern 1

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

$l_0 \ l_1 \ l_2 \ l_3 \ l_4 \ l_5 \ l_6$
Even
 even number
 print Hoja

odd
 odd
 print *
 print Hoja

Step 01: Count the numbers of row (outer loop)

Numbers of row = $n = 4$ ($row < h$)

Step 02: See what is happening in each row (inner loop)

$col < (2 * row + 1)$
 $R_0 = 1 * h$
 $R_1 = 3 * h$

Even
 pair number
 print *
 print Haja
 (row+1)

R₀ = 1ch
 R₁ = 3ch
 R₂ = 5ch
 R₃ = 7ch

CODE OF PATTERN 15

```

● ● ●

// Pattern 15: Fancy pattern 1
#include<iostream>
using namespace std;

int main(){
  int n;
  cin >> n;

  // Outer loop
  for(int row=0; row<n; row = row+1){
    for(int col=0; col<(2*row+1); col = col+1){
      if(col%2==0){
        cout << row+1 << " ";
      }
      else{
        cout << "* ";
      }
    }
    cout << endl;
  }
  return 0;
}

/*
OUTPUT: when n is 4
1
2 * 2
3 * 3 * 3
4 * 4 * 4 * 4
*/

```

PATTERN:16 Fancy pattern 2

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

(n-row) Second
 printing Hania Hai
 Even col pair

R₀ - 7ch → (2*n - 2*row - 1)
 R₁ - 5ch
 R₂ - 3ch
 R₃ - 1ch

Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 4 (row < h)

Step 02: See what is happening in each row (inner loop)

First → col < (2*row + 1)

R₀ = 1ch
 R₁ = 3ch
 R₂ = 5ch
 R₃ = 7ch

CODE OF PATTERN 16

```

// Pattern 16: Fancy pattern 2
#include<iostream>
using namespace std;

int main(){
    int num;
    cin >> num;
    int n=num/2;

    // Todo 1: First
    for(int row=0; row<n; row = row+1){
        for(int col=0; col<(2*row+1); col = col+1){
            if(col%2==0){
                cout << row+1 << " ";
            }
            else{
                cout << "* ";
            }
        }
        cout << endl;
    }

    // Todo 2: Second
    for(int row=0; row<n; row = row+1){
        for(int col=0; col<(2*n-2*row-1); col = col+1){
            if(col%2==0){
                cout << n-row << " ";
            }
            else{
                cout << "* ";
            }
        }
        cout << endl;
    }
    return 0;
}

/*
OUTPUT: when n is 8
1
2 * 2
3 * 3 * 3
4 * 4 * 4 * 4
4 * 4 * 4 * 4
3 * 3 * 3
2 * 2
1
*/

```

PATTERN:17 Hollow inverted half pyramid

R0	*	*	*	*	*	*
R1	*	-	-	-	*	
R2	*	-	-	*		
R3	*	-	*			
R4	*	*				
R5	*					
	C0	C1	C2	C3	C4	C5

Step 01: Count the numbers of row (outer loop)
 Numbers of row = n = 6 ($\text{row} < h$)

Step 02: See what is happening in each row (inner loop)
 $\text{col} < (n - \text{row})$

R0 = 6 - 1
 R1 = 5 - 1
 R2 = 4 - 1
 R3 = 3 - 1
 R4 = 2 - 1
 R5 = 1 - 1

CODE OF PATTERN 17

```

● ● ●

// Pattern 17: Inverted Hollow Half Pyramid
#include<iostream>
using namespace std;

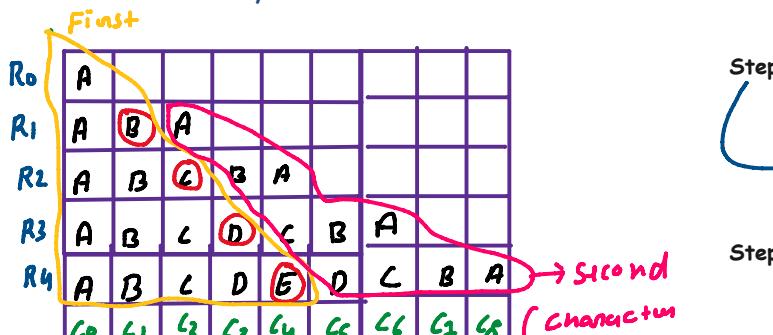
int main(){
    int n;
    cin >> n;

    // Outer loop
    for(int row=0; row<n; row = row+1){
        for(int col=0; col<n-row; col = col+1){
            if(row==0 || row==n-1){
                cout << "* ";
            }
            else{
                if(col==0 || col==(n-row-1)){
                    cout << "* ";
                }
                else{
                    cout << " ";
                }
            }
            cout << endl;
        }
        return 0;
    }

/*
OUTPUT: when n is 6
* * * * *
*   *
*   *
*   *
*   *
*/

```

PATTERN:19 Fancy Pattern 4



Step 01: Count the numbers of row (outer loop)

Numbers of row = n = 5 ($row < h$)

Step 02: See what is happening in each row (inner loop)

First

How To Print character
at Rows $ch = col + "A" + 1 - 1$

$$\begin{aligned}
 R0 &= 0 + A + 1 - 1 = A \Rightarrow 1^{st} \\
 R1 &= 1 + A + 1 - 1 = B \Rightarrow 2^{nd} \\
 R2 &= 2 + A + 1 - 1 = C \Rightarrow 3^{rd} \\
 R3 &= 3 + A + 1 - 1 = D \Rightarrow 4^{th} \\
 R4 &= 4 + A + 1 - 1 = E \Rightarrow 5^{th}
 \end{aligned}
 \left. \right\} (col < row+1)$$

CODE OF PATTERN 19

```
// Pattern 19: Fancy pattern 4
#include<iostream>
using namespace std;

int main(){
    int n;
    cin >> n;
    char ch;

    // Outer loop
    for(int row=0; row<n; row = row+1){
        // TODO 1: First
        for(int col=0; col<(row+1); col = col+1){
            int number = col+1;
            ch = number + 'A' - 1;
            cout << ch << " ";
        }
        // TODO 2: Second
        for(char alphabet=ch; alphabet>'A';){
            alphabet = alphabet - 1;
            cout << alphabet << " ";
        }
        // New line
        cout << endl;
    }
    return 0;
}

/*
OUTPUT: when n is 5
A
A B A
A B C B A
A B C D C B A
A B C D E D C B A
*/
```

HOMEWORK : 04

↳ Abni Remaining Main Hai
but jaldi prouvidi kar doonga - 😊

LinkedIn ⇒ @manojOFFICIALMj