Date: .....
Page: .....

Pattern Printing.

How to Approach :-

i) count the total no. of lines.

Total no. of lines = Total no. of rows.

Total no. of rows = Total no. of times outer loop will run.

ii) Identify what is happing in each row.

i) How many columns are there

ii) Types of elements in columns (Eg:- star, number, space, Alphabet)

iii) Try to find the formula relating rows & column.

1) Square pattern

```
* * * * ← Row 0
* * * * ← Row 1
* * * * ← Row 2
* * * * ← Row 3
col col col col
 0   1   2   3
```

Observation :-

R0 → 4 star

R1 → 4 "

R2 → 4 "

R3 → 4 "

Every row Print 4 star

Pseudo code :

```
for (i → 4)
    for (j → 4)
        Print "*"
```

2) Rectangle pattern

```
* * * * * → R0
* * * * * → R1
* * * * * → R2
↓ ↓ ↓ ↓ ↓
col col col col col
 0   1   2   3   4
```

Observation :

Every row has 5 star &

Total no. of rows = 3

Pseudo code :

```
for (i=0 → 3) {
    for (j=0 → 5) {
        cout << "*";
    }
    cout << endl;
}
```

Date: .....

Page: .....

3) Hollow Rectangle

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

Total no. of row = 4.

R0 = 5 star

R1 = 1 star 3 space 1 star

R2 = 1 " " "

R3 = 5 star.

Pseudo code :

```
for i=0; i<4; i++;
    for j=0; j<5; j++;
        if (i==0 || i==3 || j==0 || j==4)
            Print ("*");
        else
            Print (" ");
```

4) Half pyramid

```
*           → R0
* *         → R1
* * *       → R2
* * * *     → R3
```

Total no. of row = 4

R0 → 1 star

R1 → 2 "

R2 → 3 "

R3 → 4 "

} In every row (row+1) star Print.

outer loop = 0 → <=3

inner loop = 0 → row+1.

Print "*";

5) Inverted Half pyramid

```
* * * * → R0
* * *   → R1
* *     → R2
*       → R3
```

Total no. of row = 4. = n.

R0 = 4 stay

R1 = 3 stay

R2 = 2 "

R3 = 1 "

} In every row (n-row) stay Print.

outer loop = 0 → n(4)

inner loop = 0 → (n-row)

Print ("*");

6) Numeric Half pyramid

```
1             → R0
1 2           → R1
1 2 3         → R2
1 2 3 4       → R3
1 2 3 4 5     → R4
```

Total no. of row (n) = 5

R0 = 1 number = (row+1)

R1 = 2 " = "

R2 = 3 " = "

R3 = 4 " = "

R4 = 5 " = "

outer loop = 0 → 5(n)

inner loop = 0 → row+1

Print = col+1

Date: .....

Page: .....

7) Inverted Numeric Half Pyramid

```
1 2 3 4 5 → R0
1 2 3 4   → R1
1 2 3     → R2
1 2       → R3
1         → R4
```

Total no. of rows (n) = 5

R0 = 5 num (n-row)

R1 = 4 num (")

R2 = 3 " (")

R3 = 2 " (")

R4 = 1 " (")

outerloop = 0 → 5

innerloop = 0 → (n-row)

Print = col + 1

8) Full pyramid

```
- - - - *       → R0
- - - * * *     → R1
- - * * * * *   → R2
- * * * * * * * → R3
* * * * * * * * * → R4
```

Total no. of rows (n) = 5

R0 = 4 space 1 star

R1 = 3 " 2 "

R2 = 2 " 3 "

R3 = 1 " 4 "

R4 = 0 " 5 "

(n-row-1) space, (row+1) star

Outerloop = 0 → n

innerloop1 = 0 → n-row-1

Print (" ")

innerloop2 = 0 → row+1

Print ("*").

9)

```
* * * * R0   * * * *
- * * *  R1   - * * *
- - * *  R2   - - * *
- - - *  R3   - - - *
```

→ Inverted Full Pyramid.

Total no. of rows (n) = 4

R0 = 0 space 4 star

R1 = 1 " 3 "

R2 = 2 " 2 "

R3 = 3 " 1 "

no. of row space, n-row star

outer loop = 0 → n

innerloop1 = 0 → row

Print (" ")

innerloop2 = 0 → n-row

Print ("*")

Basics of Programming - Level 2
Pattern continues.

Date:
Page:

10)

Diamond Pattern

```
- - - *      R0
- - * *      R1
- * * *      R2
* - * - *    R3
* * * *      R0
- * * *      R1
- - * *      R2
- - - *      R3
```

Total no. of rows (n) = 8

For upper half.

Total no. of row (n) = 4.

R_0 = 3 space, 1 star
R_1 = 2 " , 2 "
R_2 = 1 " , 3 "
R_3 = 0 " , 4 "

(n-row-1) space, (row+1) star

For lower half

Total no. of row (n) = 4

R_0 = 4 star, 0 space
R_1 = 3 " , 1 "
R_2 = 2 " , 2 "
R_3 = 1 " , 3 "

(n-row) (star), (no. of row) (space)

⇒

```
outerloop : 0 → n (4)
    Innerloop1 : 0 → no. of row
        Print (" ")
    Innerloop 2 : 0 → n - row
        Print ("*")
```

For upper half

```
outerloop = 0 → n (4)
    Innerloop1 : 0 → n-row-1
        Print (" ")
    Inner loop 2 : 0 → row+1
        Print("*")
```

Date: .....
Page: .....

11) **Hollow Pyramid**

```
- - - *        R0
- - * - *      R1
- * - - - *    R2
* - - - - - *  R3
```

Total no. of rows (n) = 4.

R0 = 3 space, 1 star
R1 = 2 " , 2 "
R2 = 1 " , 2 "
R3 = 0 " , 2 "

(n-row-1) (row+1)

∴ Here, we're printing star on 1st & last column of every row.

outer loop = 0 → n
  Inner loop 1 = 0 → n-row-1 & print (" ")
  Inner loop 2 = 0 → row + 1. if (col == 0 || col = row+1-1)
    Print star for 1st & last col → print ("*")
  Print space betn 1st & last col → else, print (" ")

12) **Inverted Hollow Pyramid**

```
R0 * - - - - - *
R1 - * - - - *
R2 - - * - *
R3 - - - *
```

Total no. of row (n) = 4.

R0 = 2 star, 5 space, 0 space (space
R1 = 2 " , 3 " , 1 " before
R2 = 2 " , 1 " , 2 " star)
R3 = 1 " , 0 " , 3 "

(n-row) no. of row (row)

∴ Here, for every row, Print star for 1st & last column

Outer loop = 0 → n
Inner loop 1 = 0 → row, print (" ") → space before star.
Inner loop 2 = 0 → n-row, if (col == 0 || col = n-row-1)
  Print ("*")
  else print (" ")

Date: .....

Page: .....

→ **Inverted Hollow Pyramid**

```
R0  *-----*      R0 = 0 space, 7 character (star + space)
R1  -*---*       R1 = 1 ",  5 "  ( "  )
R2  --*-*        R2 = 2 ",  3 "  ( "  )
R3  ---*         R3 = 3 ",  1 "  ( "  )
```

no. of row (row) — (2n-2r-1) — { n = Total no. of row, r = current row

```
OuterLoop = 0 → n        row
InnerLoop1 = 0 →
    Print (" ")  → print space before star
InnerLoop2 = 0 → 2n-2r-1
    if (col == 0 || col = n-row-1)
        Print ("*")      ↳ Last column of every row
    else Print (" ")
```

13)

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

Upper Half
⇒ Hollow Pyramid
↳ use the code of Hollow Pyramid

Lower Half
⇒ Inverted Hollow Pyramid
↳ use the code of Inverted Hollow Pyramid

14)

```
1              R0
2 * 2          R1
3 * 3 * 3      R2
4 * 4 * 4 * 4  R3
c0 c1 c2 c3 c4 c5 c6
```

Total no. of row (n) = 4
R0 = 1 character (2r+1)
R1 = 3 " (")
R2 = 5 " (")
R3 = 7 " (")

Odd column = '*' → print
Even " = 'row+1' → print

Date: .....
Page: .....

```
outerloop : 0 → n
Innerloop : 0 → 2r+1
        if (col % 2 == 1)  print ("*");
        else  print (row+1);
```

Soln No. 1

```
outerloop : 0 → n
  Innerloop : 0 → row+1  → Total no. of col
    if (col == row+1-1)  → Last column of every row.
          Print (row+1)
    else  print (row+1) ¢ ("*")
```

Soln No. 2

15) Hollow Inverted Half Pyramid

```
R0  * * * * * *
R1  * - - - *
R2  * - - *
R3  * - *
R4  * *
R5  *
```

Total no. of row (n) : 6

observation :-

1) In 1st ¢ last row has no space
   ∴ row == 0 || row == n-1
   Print ("*")

2) For other row, only 1st ¢ last column has printed star
   ∴ col == 0 || col == n-row-1
   Print ("*")

otherwise, print (" ").

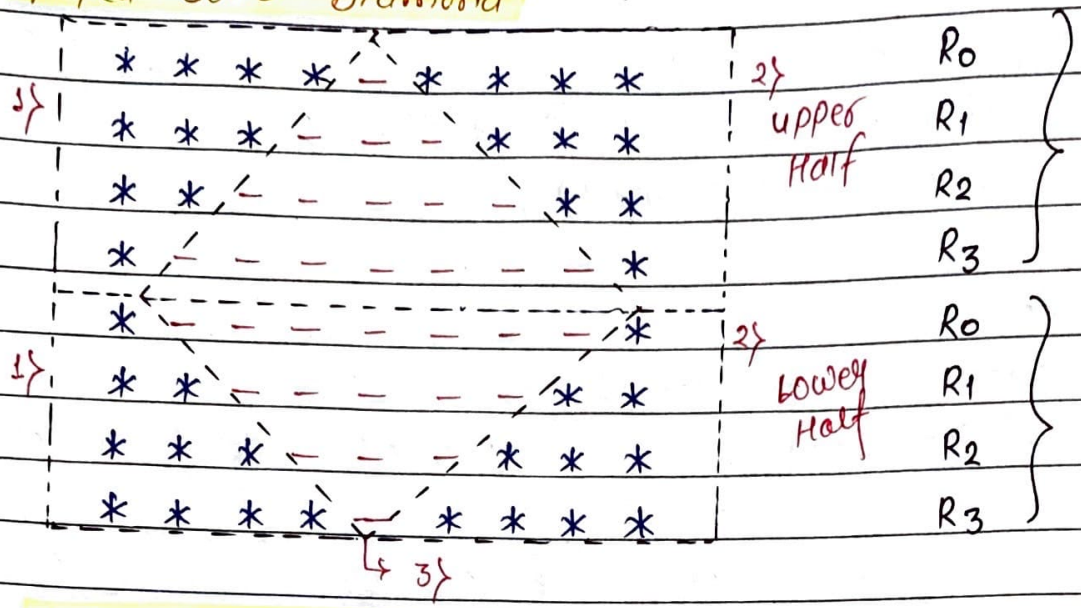
```
outerloop : 0 → n
  Innerloop : 0 → n-row
    ~~if (col==0 || col==n-row-1)~~
```

1) if (row == 0 || row == n-1)
       print ("*");
   else

2) if (col == 0 || col == n-row-1)
       Print ("*")
    else print (" ").

Date: .....
Page: .....

16) **Fliped Solid Diamond**



→ **observation for upper Half**

R_0 = 4 star	Inverted Half Pyramid.		R_0 = 1 space	Full Pyramid of space.	
R_1 = 3 star			R_1 = 3 "		
R_2 = 2 star			R_2 = 5 "		
R_3 = 1 "			R_3 = 7 "		
(n-row)			(2r+1)		

R_0 = 4 star	Inverted Half Pyramin
R_1 = 3 "	
R_2 = 2 "	
R_3 = 1 "	
(n-row)	

Total no. of row (n) : 4

Outerloop : 0 → n

Innerloop 1 : 0 → n-row, print (" * ")

Innerloop 2 : 0 → 2r+1, print (" ") r = row

Innerloop 3 : 0 → n-row, print (" * ")

Date: .....

Page: .....

→ observation for Lower Half

R_0 = 1 Star	} Half Pyramid		R_0 = 7 Space	} Full Pyramid of Space	
R_1 = 2 Star			R_1 = 5 "		
R_2 = 3 "			R_2 = 3 "		
R_3 = 4 "			R_3 = 1 "		
(row+1)			(2n-2r-1)		

R_0 = 1 Star
R_1 = 2 "
R_2 = 3 "
R_3 = 4 "
(row+1)
} Half pyra mid

Total no. of row (n) = 4.

outerloop = 0 → n

Inner loop 1 : 0 → row+1, Print("*")

Inner loop 2 : 0 → 2n-2r-1, Print(" ")

Inner loop 3 : 0 → row+1, Print("*")

17)

1
2 * 2
3 * 3 * 3
4 * 4 * 4 * 4
} upper Half

R_0
R_1
R_2
R_3
} Total no. of row (n) : 4

3 * 3 * 3
2 * 2
1
} Lower Half

R_0
R_1
R_2
} Total no. of row (n) : 3

→ upper Half

⇒ use the same code of pattern No. 14).

→ Lower Half

R_0 = 5 charcter
R_1 = 3 "
R_2 = 1 "
(2n-2r-1)

Total no. of row (n) : 3

outerloop : 0 → n

Innerloop : 0 → 2n-2r-1

if (col % 2 == 1) Print("*")

else Print(n-row) & (" ")

Date: .....
Page: .....

18)

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

Total no. of row (n) = 6

observation :

i) In 1st & last row has no space

ii) except 1st & last row, every row has some spaces & numbers.

R0 = 1 character } (row+1) character print.
R1 = 3 "
R2 = 4 " } (row+2) character print
R3 = 5 "
R4 = 6 "
R5 = 6 " } n character print

```
if (row == 0)  print (1);
if (row = n-1)  print (1 ~ (row+1));
if (row != 0 && row != n-1)
        print (1-2) (1--3) (1---4) ----

outerloop = 0 → n
    if (row == 0)  print (row+1);
    else if (row == n-1), Then
        Innerloop1 = 0 → n , print (col+1);
    else
        Innerloop2 = 0 → row+2
            if (col == 0)  print (col+1);
            else if (col == row+2-1)  print (col);
            else  print (" ");
```

Date: .....
Page: .....

19) **Numeric Hollow Half Pyramid.**

```
R0   1
R1   1 2
R2   1 - 3
R3   1 - - 4
R4   1 2 3 4 5
     C0 C1 C2 C3 C4
```

Total no. of rows (n) = 5

Outer loop = 0 → n (row < n)

Inner loop = 0 → row (col <= row)

if (col == 0 || col == row || row == n-1) (Last row)

print (col+1);

else print (" ");

20)

```
A          R0
A B A      R1
A B C B A  R2
A B C D C B A  R3
A B C D E D C B A  R4.
```

Total no. of rows (n) = 5

```
Outer loop = 0 → n;
    char ch;
Inner loop1 = 0 → row+1
    int num = col+1;
    ch = num + 'A' - 1;   // mapping of character
    Print (ch);

Inner loop2 = ch → 'A'
Print character until reach to 'A'
for (char alpha = ch; alpha > 'A'; ) {
        alpha --;
        print (alpha);
}
cout << endl;
```

This loop print Triangle of character

This loop Print character till 'A'.

Date: .....

Page: .....

21) **Half pyramid with number**

```
1                R0
1 2              R1
1 2 3            R2
1 2 3 4          R3
1 2 3 4 5        R4
```

Total no. of row (n) : 5

outerloop : 0 → n

Innerloop : 0 → row+1

print (col+1);

22) **Inverted Half pyramid with number**

```
1 2 3 4 5        R0
1 2 3 4          R1
1 2 3            R2
1 2              R3
1                R4
```

Total no. of rows (n) : 5

outerloop : 0 → n

Innerloop : 0 → n - row+1

print (col+1);

23)

```
1                R0
2 2              R1
3 3 3            R2
4 4 4 4          R3
5 5 5 5 5        R4
```

Total no. of rows (n) : 5

outerloop : 0 → n

Innerloop : 0 → row+1

print (row+1);

24) **Floyd's Triangle**

```
1                R0
2 3              R1
4 5 6            R2
7 8 9 10         R3
11 12 13 14 15   R4
```

Total no. of rows (n) : 5

int num = 1;

outerloop : 0 → n

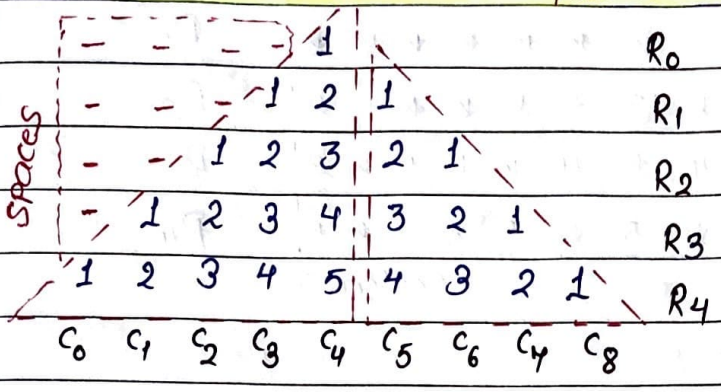
Innerloop : 0 → row+1

print (num);

num ++;

Date: .....
Page: .....

25) **Numeric Palindrome Equilatral Pyramid**



Total no. of row(n) : 5

observation :-

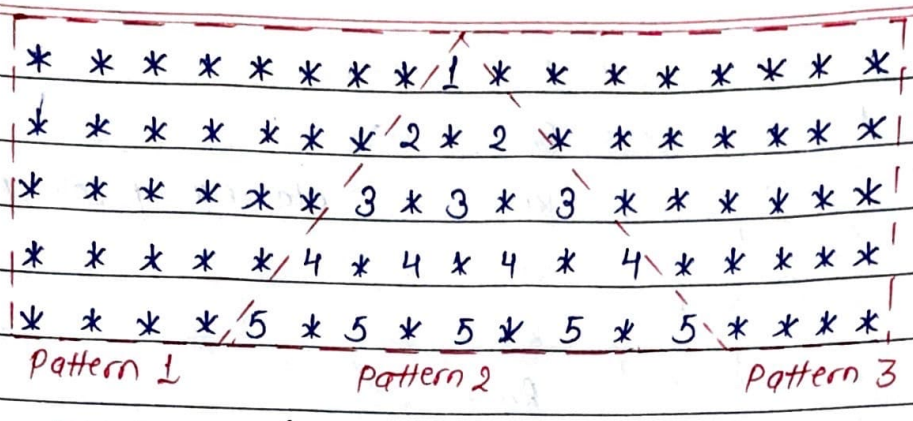
R_0 =	4 space,	1 Number	R_0 =	0 Number	
R_1 =	3 " ,	2 "	R_1 =	1 "	
R_2 =	2 " ,	3 "	R_2 =	2 "	
R_3 =	1 " ,	4 "	R_3 =	3 "	
R_4 =	0 " ,	5 "	R_4 =	4 "	
	(n-row-1)	(row+1)		(row)	
	Left Triangle			Right Triangle	

```
Outerloop : 0 → n
    Innerloop1 : 0 → n-row-1 , Print(" "); (Space Before num).
      int num = 1;
    Innerloop2 : 0 → row+1,
                 Print(num);
                 num ++;          } Print left hand side number Triangle.

    Innerloop3 : num-1 → 1        } Print Right hand side number Triangle till '1'.
                 Print(col)
    for(int col= num-1; col>1;){
        col--;  Print(col);
    }
```

Date: .....
Page: .....

26)



Total no. of row (n) = 5

observation for pattern 1 & 3

R0 = 8 stay
R1 = 7 "
R2 = 6 "
R3 = 5 "
R4 = 4 "

(2n - row - 2)

observation for Pattern 2.

R0 = 1 character
R1 = 3 "
R2 = 5 "
R3 = 7 "
R4 = 9 "

(2 * row + 1)

odd column = '*' print
Even " = 'row+1' print.

Outerloop = 0 → n

Innerloop 1 = 0 → 2n - row - 2
print ("*");
For print stay

Innerloop 2 = 0 → 2 x row + 1.
if (col % 2 == 1) print ("*");
else print (row+1);
For print num & stay

Innerloop 3 = 0 → 2n - row - 2
print ("*");
For print stay.

Date: .....

Page: .....

27) Solid Half Diamond

```
*          R0
*  *       R1
*  *  *    R2
*  *  *  *  R3
-----------
*  *  *    R0
*  *       R1
*           R2
```

Total no. of row (n) = 4.
⇒ (row+1) star Print

Total no. of row (n) : 3
R0 = 3 star
R1 = 2 "
R0 = 1 "
} (n-row-1)

Upper Half

```
outerloop : 0 → n
    Innerloop : 0 → row+1
        Print ("*");
```

Lower Half

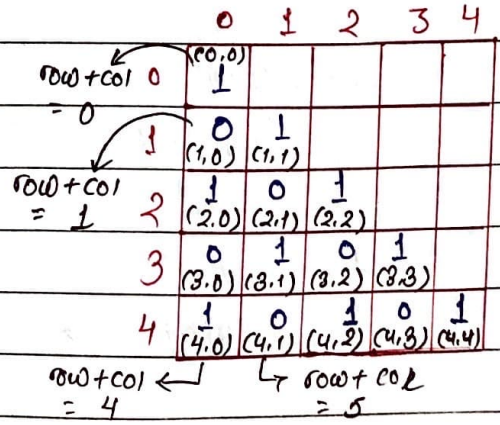
```
outerloop : 0 → n
    Innerloop : 0 → n-row-1.
        Print ("*");
```

28) 0 - 1 triangle

```
1              R0
0  1           R1
1  0  1        R2
0  1  0  1     R3
1  0  1  0  1  R4.
```

Total no. of rows (n) = 5
R0 = 1 number
R1 = 2 "
R2 = 3 "
R3 = 4 "
R4 = 5 "
} (row+1).

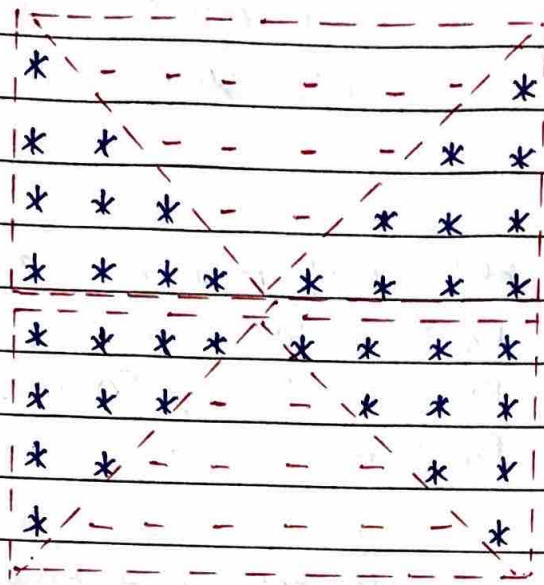
```
outerloop : 0 → n
    Innerloop : 0 → row+1
        int sum = row + col;
        if (sum % 2 == 0)
            Print ("1 ");
        else
            print ("0 ");
```



When row+col = Even Then Print 1, otherwise Print 0.

Date: .....
Page: .....

29) Butterfly Pattern



Observation for upper Half

	Star	Space	Star
R_0 =	1 Star	6 Space	1 Star
R_1 =	2 "	4 "	2 "
R_2 =	3 "	2 "	3 "
R_3 =	4 "	0 "	4 "
	(row+1)	(2n-2row-2)	(row+1)

```
Outerloop = 0 ⟶ n        Total no. of rows (n) = 4
    Innerloop1 : 0 ⟶ row+1
        Print ("*");

    Innerloop2 : 0 ⟶ (2n-2row-2)
        Print (" ");

    Innerloop3 : 0 ⟶ row+1
        Print ("*");
```

Date: .....
Page: .....

Observation for Lower Half

	Star	Space	Star
R_0 =	4 Star	0 Space	4 Star
R_1 =	3 "	2 "	3 "
R_2 =	2 "	4 "	2 "
R_3 =	1 "	6 "	1 "
	(n-row)	(2*row)	(n-row)

Total no. of rows (n) = 4

```
outerloop = 0 → n.
  Innerloop1 = 0 → (n-row)
      Print ("*");

  Innerloop2 = 0 → (2xrow)
      Print (" ");

  Innerloop3 = 0 → (n-row)
      Print ("*");
```

30) Numeric Hollow Inverted Half Pyramid

```
1 2 3 4 5   R0
2 - - 5     R1
3 - 5       R2
4 5         R3
5           R4
C0 C1 C2 C3 C4
```

Total no. of row (n) = 5

```
outerloop = 0 → n
  Innerloop = row+1 → <=n
    if (col == row+1 || col == n ||
        row == 0)
      Print (col);
    else
      Print (" ");
```