

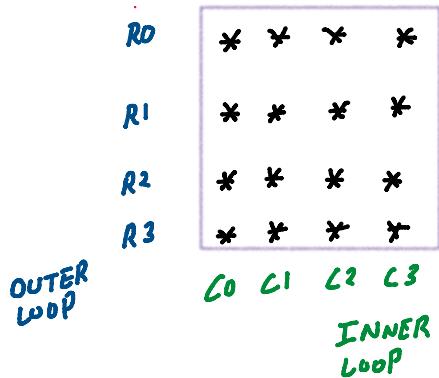
HW CLASS:03 — 28 / 08 / 23
<https://www.linkedin.com/in/manojofficialmj/>

PATTERN PROBLEMS

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

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

PATTERN:01 Square pattern



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

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

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

($\text{Col} < 4$)

Row0 = print 4 star
Row1 = print 4 star
Row2 = print 4 star
Row3 = print 4 star

CODE OF PATTERN 01

```

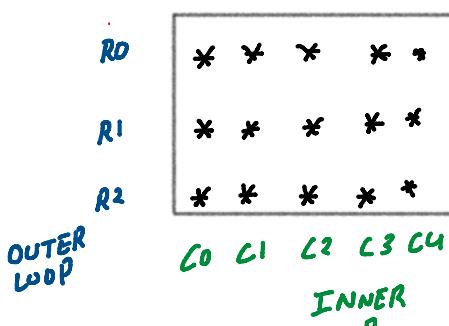
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 4;
6
7     // outer Loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to n-1
10        for(int col = 0; col < n; col = col + 1){
11            cout << "* ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }
```

DRY RUN OF PATTERN 01

n 4

row	row < n	col	col < n	point *	col = col + 1	next line	row = row + 1
0	0 < 4	0	0 < 4	*	1		
		1	1 < 4	*	2		
		2	2 < 4	*	3		
		3	3 < 4	*	4		
		4	4 < 4	-	-		✓ 1
1	1 < 4	0	0 < 4	*	1		
		1	1 < 4	*	2		
		2	2 < 4	*	3		
		3	3 < 4	*	4		
		4	4 < 4	-	-		✓ 2
2	2 < 4	0	0 < 4	*	1		
		1	1 < 4	*	2		
		2	2 < 4	*	3		
		3	3 < 4	*	4		
		4	4 < 4	-	-		✓ 3
3	3 < 4	0	0 < 4	*	1		
		1	1 < 4	*	2		
		2	2 < 4	*	3		
		3	3 < 4	*	4		
		4	4 < 4	-	-		✓ 4
4	4 < 4	0	0 < 4	*	1		
		1	1 < 4	*	2		
		2	2 < 4	*	3		
		3	3 < 4	*	4		
		4	4 < 4	-	-		✓ Outer loop ending

PATTERN:02 Rectangle pattern



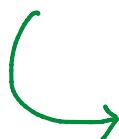
Step 01: Count the numbers of row (outer loop)

Numbers of row = 3 (row < 3)

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

OUTER LOOP C₀ C₁ C₂ C₃ C₄
 INNER LOOP

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



Row0 = print 5 star
 Row1 = print 5 star
 Row2 = print 5 star

(C₀₁ < (3 + 2))

CODE OF PATTERN 02

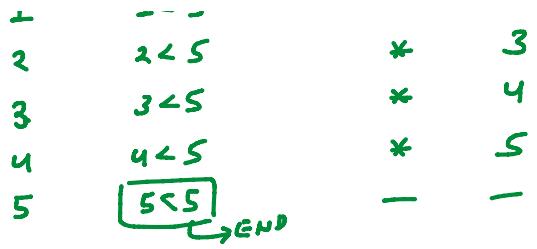
```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 3;
6
7     // outer loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to n+1
10        for(int col = 0; col < (n+2); col = col + 1){
11            cout << "* ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }
```

DRY RUN OF PATTERN 02

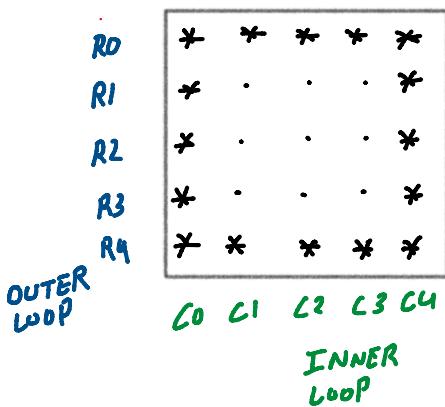
n [3]

row	row < n	col	col < (n+2)	Print *	Col = Col + 1	Next Row	row < n + 1
0	0 < 3	0	0 < 5	*	1		
		1	1 < 5	*	2		
		2	2 < 5	*	3		
		3	3 < 5	*	4		
		4	4 < 5	*	5		
		5	5 < 5	—	—		1
1	1 < 3	0	0 < 5	*	1		
		1	1 < 5	*	2		
		2	2 < 5	*	3		
		3	3 < 5	*	4		
		4	4 < 5	*	5		
		5	5 < 5	—	—		2
2	2 < 3	0	0 < 5	*	1		
		1	1 < 5	*	2		
		2	2 < 5	*	3		
		3	3 < 5	*	4		



3
3 < 3
 outer loop ending

PATTERN:03 Hollow rectangle pattern



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

Numbers of row = 5 (ROW < 5)

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

Row0 = print 5 star
 Row1 = print 1* 3space 1*
 Row2 = print 1* 3space 1*
 Row3 = print 1* 3space 1*
 Row4 = print 5 star

if (no || 0) L print 5 times *

if (c0 || c4) L print 1 time *

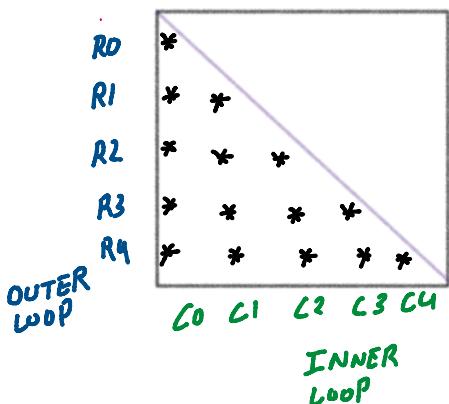
else L print 3 times space

CODE OF PATTERN 03

```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 5;
6
7     // outer loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner loop traverse from 0 to n-1
10        for(int col = 0; col < n; col = col + 1){
11            // if zeroth or last row, then print n times star
12            if(row == 0 || row == n-1){
13                cout << "* ";
14            }
15            else{
16                // if zeroth or last col, then print 1 time star
17                if(col == 0 || col == n-1){
18                    cout << "* ";
19                }
20                else{
21                    cout << " ";
22                }
23            }
24        }
25        cout << endl;
26    }
27    return 0;
28 }
```

PATTERN:04 Half Pyramid pattern



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

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

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

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

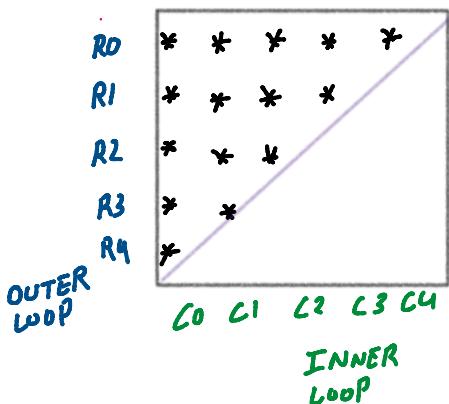
($\text{Col} < \text{Row} + 1$)

CODE OF PATTERN 04

```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 5;
6
7     // outer Loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to (row+1)-1
10        for(int col = 0; col < (row + 1); col = col + 1){
11            cout << "* ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }
```

PATTERN:05 Inverted half pyramid pattern



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

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

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

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

($\text{Col} < (5 - \text{Row})$)

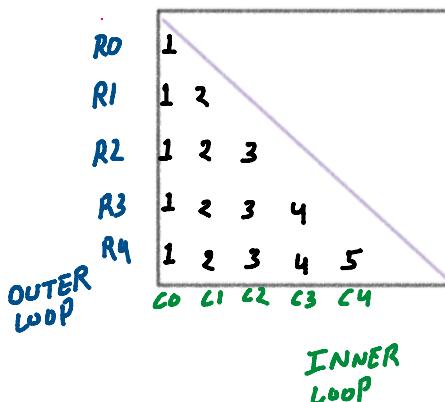
CODE OF PATTERN 05

```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 5;
6
7     // outer Loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to n-row
10        for(int col = 0; col < (n - row); col = col + 1){
11            cout << "* ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }

```

PATTERN:06 Numerical half pyramid pattern



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

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

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

$Row0 = \text{print } 1$
 $Row1 = \text{print } 1-2$
 $Row2 = \text{print } 1-3$
 $Row3 = \text{print } 1-4$
 $Row4 = \text{print } 1-5$

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

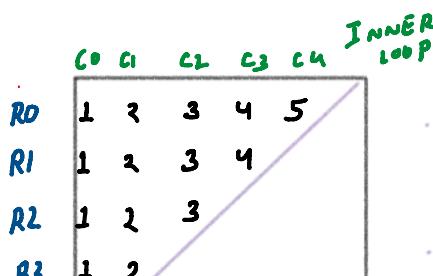
CODE OF PATTERN 06

```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 5;
6
7     // outer Loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to (row+1)-1 and print col+1
10        for(int col = 0; col < (row + 1); col = col + 1){
11            cout << col+1 << " ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }

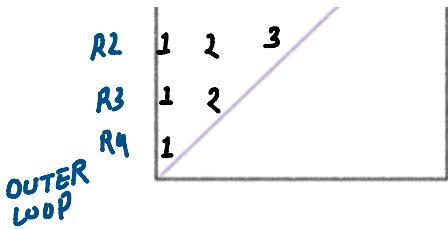
```

PATTERN:07 Inverted numerical half pyramid pattern



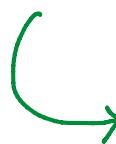
Step 01: Count the numbers of row (outer loop)

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



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

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



Row0 = print 1-5
Row1 = print 1-4
Row2 = print 1-3
Row3 = print 1-2
Row4 = print 1

($col < (5 - row)$)

CODE OF PATTERN 07

```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int n = 5;
6
7     // outer Loop traverse from 0 to n-1
8     for(int row = 0; row < n; row = row + 1){
9         // inner Loop traverse from 0 to n-row and print col+1
10        for(int col = 0; col < (n-row); col = col +1){
11            cout << col+1 << " ";
12        }
13        cout << endl;
14    }
15    return 0;
16 }
```

All flowcharts are converted into C++ programs

```

// Program01: Multiply two numbers by taking input from user
#include<iostream>
using namespace std;

int main(){
    int a, b;
    cout << "Enter a" << endl;
    cin >> a;
    cout << "Enter b" << endl;
    cin >> b;
    int result = a * b;
    cout << "Multiply of a and b is " << result << endl;
    return 0;
}

/*
OUTPUT:
Enter a
2
Enter b
3
Multiply of a and b is 6
*/

```

LinkedIn: @manojofficialmj

```

// Program02: Find the perimeter of a triangle
#include<iostream>
using namespace std;

int main(){
    int a = 3, b = 4, c = 5;
    int p = a + b + c;
    cout << "Perimeter of a triangle is " << p << endl;
    return 0;
}

/*
OUTPUT:
Perimeter of a triangle is 12
*/

```

LinkedIn: @manojofficialmj

```
● ● ●
// Program03: Find the simple interest
#include<iostream>
using namespace std;

int main(){
    int p = 100, r = 2, t = 1;
    double SI = (p*r*t)/100;
    cout << "Simple interest is " << SI << endl;
    return 0;
}

/*
OUTPUT:
Simple interest is 2
*/
Linkedin: @manojofficialmj
```

```
● ● ●
// Program04: Find the compound interest
#include <bits/stdc++.h>
#include<iostream>
using namespace std;

int main()
{
    double p = 10000, r = 5, t = 2;
    double A = p * (pow((1 + r / 100), t));
    double CI = A - p;
    cout << "Compound interest is " << CI;
    return 0;
}

/*
OUTPUT:
Compound interest is 1025
*/
Linkedin: @manojofficialmj
```

```
● ● ●
// Program05: Print counting from n to 1
#include<iostream>
using namespace std;

int main(){
    int n, counting = 0;
    cout << "Enter a number" << endl;
    cin >> n;
    for(int i = n; i > 0; i = i - 1){
        counting = counting + i;
    }
    cout << "Counting is " << counting << endl;
    return 0;
}

/*
OUTPUT:
Enter a number
3
Counting is 6
*/
Linkedin: @manojofficialmj
```

```
● ● ●
// Program06: Find the factorial of a number
#include<iostream>
using namespace std;

int main(){
    int n, fac = 1;
    cout << "Enter a number " << endl;
    cin >> n;
    if(n < 0){
        cout << "Invalid input" << endl;
    }
    else if(n == 0){
        cout << "Factorial of " << n << "! is " << fac << endl;
    }
    else if(n > 0){
        for( n ; n > 0; n = n - 1 ){
            fac = fac * n;
        }
        cout << "Factorial of " << n << "! is " << fac << endl;
    }
    return 0;
}

/*
OUTPUT:
Enter a number
-1
Invalid input

Enter a number
0
Factorial of 0! is 1

Enter a number
5
Factorial of 5! is 120
*/
Linkedin: @manojofficialmj
```

```
// Program07: Check if number is prime or not
#include<iostream>
using namespace std;

int main(){
    int n;
    cout << "Enter a number" << endl;
    cin >> n;

    if(n<2){
        cout << n << " is not prime number" << endl;
    }
    else{
        for(int i = 2; i < n; i = i + 1){
            if(n % i == 0){
                cout << n << " is not prime number" << endl;
            }
        }
        cout << n << " is prime number" << endl;
    }
    return 0;
}

/*
OUTPUT:
Enter a number
-1
-1 is not prime number

Enter a number
0
0 is not prime number

Enter a number
1
1 is not prime number

Enter a number
2
2 is prime number
*/

```

LinkedIn: @manojofficialmj

```
// Program08: Check valid triangle or not
#include<iostream>
using namespace std;

int main(){
    int a,b,c;

    cout << "Enter side a = ";
    cin >> a;
    cout << "Enter side b = ";
    cin >> b;
    cout << "Enter side c = ";
    cin >> c;

    if(a+b > c && a+c > b && b+c > a){
        cout << "Valid triangle" << endl;
    }
    else{
        cout << "Invalid triangle" << endl;
    }
    return 0;
}

/*
OUTPUT:
Enter side a = 3
Enter side b = 4
Enter side c = 5
Valid triangle

Enter side a = 0
Enter side b = 2
Enter side c = 1
Invalid triangle
*/

```

LinkedIn: @manojofficialmj

```
// Program09: Print max of three numbers
#include<iostream>
using namespace std;

int main(){
    int a,b,c;
    cout << "Enter number a = ";
    cin >> a;
    cout << "Enter number b = ";
    cin >> b;
    cout << "Enter number c = ";
    cin >> c;

    if(a>b && a>c){
        cout << "Maximum number is a" << endl;
    }
    else if(b>a && b>c){
        cout << "Maximum number is b" << endl;
    }
    else{
        cout << "Maximum number is c" << endl;
    }
}

/*
OUTPUT:
Enter number a = 1
Enter number b = 2
Enter number c = 3
Maximum number is c
*/

```

LinkedIn: @manojofficialmj