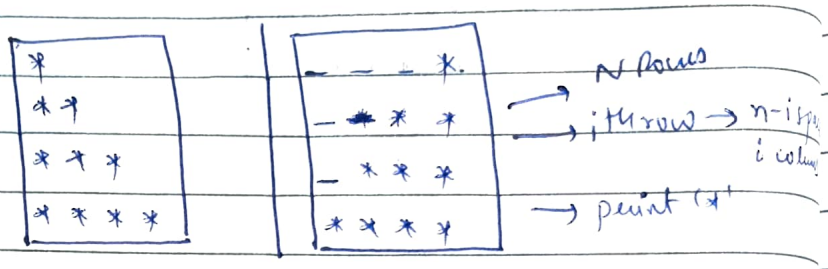
Dt 11/12/2022
Pg

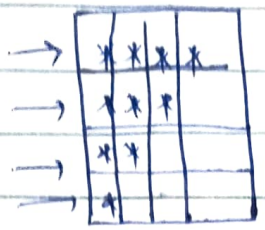
Pattern - 2 Lecture

1) Mirror Image Pattern (Reverse Triangle)




```
{
    int n;
    cin>>n;
    int i=1;
    while(i<=n){
        int space=1;
        while(space<=n-i){
            cout<<" ";
            space++;}
        int star=1;
        while(star<=i){
            cout<<"*";
            star++;}
        cout<<endl;
        i++;
    }
```

(2) Inverted Triangle (▽)



→ N Rows
→ ith Row → n-i+1 columns
→ print '*'

```
{
    int n;
    cin>>n;
    int i=1;
    while(i<=n){
        int j=1;
        while(j<=n-i+1){
            cout<<" *";
            j++;
        }
        cout<<endl;
        i++;
    }
}
```

(3) Isosceles Triangle

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

3 parts
→ Spaces
→ text increasing
→ text decreasing

```
int n;
cin>>n;
int i=1;
while(i<=n){
    int space=1;
    while(space<=n-i){
        cout<<" ";
        space++;
    }
    int j=1;
    while(j<=i){
        cout<<j;
        j++;
    }
    j=i-1;
    while(j>=1){
        cout<<j;
        j--;
    }
    cout<<endl;
    i++;
}
```

WOW!

(4) Triangle of Numbers

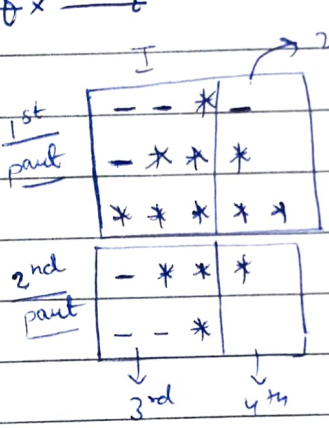
-	-	-	1			
-	-	2	3	2		
-	3	4	5	4	3	
4	5	6	7	6	5	4

3 parts → Spaces → 1 to n-i

numbers → ith row → i columns

numbers → start 2*i-2 → i-1

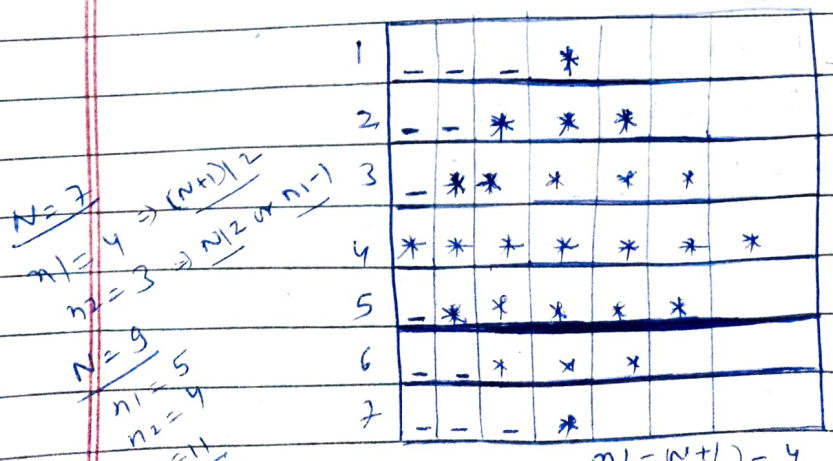
(5) Diamond of Stars



1st part → Space & star

2nd part →

→ Explanation of Diamond of Stars Problem



RowNo	No. of spaces	No. of
1	3	1
2	2	3
3	1	5
4	0	7
5	1	5
6	2	3
7	3	1

4 1st half

3 2nd half

n2 = N/2 or n1-1 = 3

WOW!

Dt
Pg

"For First Part"

Spaces → n1-i spaces → 0

Stars → 2*i-1 (i: 1 to ^)

For second part

3 Spaces → n2-i+1 spaces

2 Stars → 2i-1

1

umbering

Hit and Trial Method for Pattern Printing

Rows	Spaces	Stars
1	2	1
2	1	3
3	0	5
4	1	3
5	2	1

1st part

Spaces = n1 - i to 0

Stars = 2*i - 1

2nd part

Spaces = n2 - i + 1 to 0

Stars = 2*i - 1

n1 = (5+1)/2 = (n+1)/2 = 3

n2 = 5/2 = 2

1

2

2+1

2

	(1)	(2)	(3)	(4)	(5)
(1) →	5	5	5	5	5
(2) →	4	5	5	5	5
(3) →	3	4	5	5	5
(4) →	2	3	4	5	5
(5) →	1	2	3	4	5

ans = n-i+1

if(ans<=n){ count=ans; }

else{ count=n; }

(1) →	1	2	3	4	4	3	2	1
(2) →	1	2	3	*	*	3	2	1
(3) →	1	2	*	*	*	*	2	1
(4) →	1	*	*	*	*	*	*	1

1	2	3	4	5
16	-	-	-	6
15	-	-	-	7
14	-	-	-	8
13	12	11	10	9

1 →	____1	1st
2 →	___123	
3 →	__12345	
4 →	_1234567	
5 →	123456789	
6 →	-12345678	
7 →	--123456	
8 →	___123	
9 →	____1	

n1 = (n+1)/2 = 10/2 = 5

n2 = n - n1 = 4

1st part

space → n1-i

num → 2*i-1

2nd part

i = n2; i>=1; i--

space = n2-i+1;

num = 1 to 2*i-1

Dt.
Pg.

(5)

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

```
i=1, k=1
m[7][7]={0};
while(i<=7){
    int j=1;
    while(j<=7){
        if(j==i || 8-i==j){
            m[i-1][j-1]=k;
            j++; }
        if(i<4){
            k++;}
        else --k;
        i++; }
    i=0;
    while(i<7){
        int j=0;
        while(j<7){
            if(m[i][j]==0) cout<<" ";
            else cout<<m[i][j];
            j++; }
        i++;
        cout<<endl;
    }
}
```

END OF TOPIC