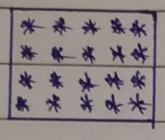
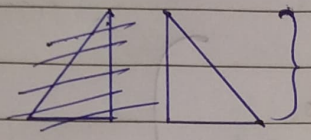
Pattern Printing.

⇒ Print The Given Patterns.



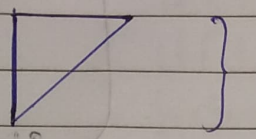
```
for(i = 1 to n)
for(j = 1 to n)
```

j ⇒ 1, 2, 3, 4, 5 ....
2*j-1 ⇒ 1, 3, 5, 7, 9 ....



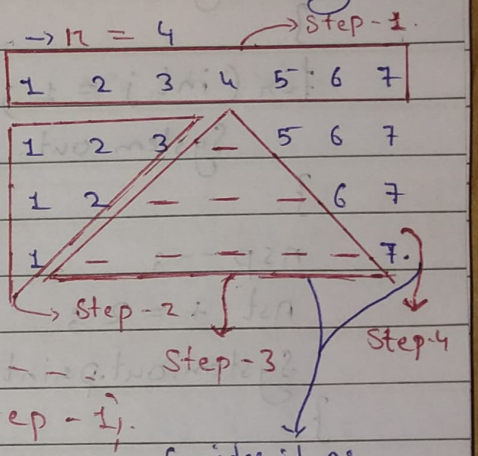
```
for(i = 1 to n)
for(j = 1 to i)
```

⇒ Composite Pattern



```
for(i = 1 to n)
for(j = 1 to n+1-i)
```

(1) Number Bridge

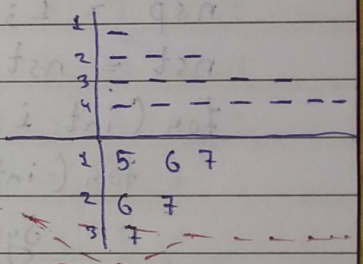


⇒ Code :-

```
Psvm {
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
for(int i = 1; i<=2*n-1; i++){
    System.out.print(i+" ");
}. System.out.println();
n--;
int nsp = 1;
for(int i = 1; i<=n; i++){ //Rows.
    int a = 1;
    for(int j = 1; j<=n+1-i; j++){ //Numbers.
        System.out.print(a++ + " ");
    }
    for(int j = 1; j<=nsp; j++){ //Spaces.
        System.out.print(" " + " ");
        a++; // Very Imp.
    }
    nsp += 2;
    for(int j = 5; j<=n+5-i; j++){ //Numbers.
        System.out.print(a++ + " ");
    }.
    System.out.println();
    a += 1;
}.
}.
```

Step-1.

Consider it as.



Step-2.

Step-3

Step-4.

nsp ⇒ No. of Spaces.
nst ⇒ No. of Stars.

(2) Star Diamond :-

⇒ Take "n" as a input :

M T W T F S S | Page No.: | Date: | YOUVA

⇒ Code :- Psvm {

```
int nsp = n-1;
int nst = 1;
for(int i=1; i<=n; i++){ //Rows
    for(int j=1; j<=nsp; j++){ //Spaces
        System.out.print(" ");
    }
    for(int j=1; j<=nst; j++){ //Stars
        System.out.print("*" + " ");
    }
    nsp--;
    nst+=2;
    System.out.println();
}
nsp = 1;
nst = nst-4;
for(int i=1; i<=n-1; i++){ //Rows.
    for(int j=1; j<=nsp; j++){ //Spaces.
        System.out.print(" ");
    }
    for(int j=1; j<=nst; j++){ //Stars.
        System.out.print("*" + " ");
    }
    nsp++;
    nst-=2;
    System.out.println();
}
```

Step-1.

Step-2

n = 4

```
1 - - - *
2 - - * * *
3 - * * * * *
4 * * * * * * *
1 - * * * * *
2 - - * * * *
3 - - - *
```

Explanation

Step 1.

nsp = n-1 , nsp--

nst = 1 , nst+=2

Step 2.

nsp = 1 , nsp++

nst = nst+4 , nst-=2.

Toughest Question In pattern Printing's

Page No.: YOUVA
Date:

(3) Spiral Number :-

⇒ Code :- Psvm {

```
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
for (int i = 1; i <= 2*n-1; i++) {
    for (int j = 1; j <= 2*n-1; j++) {
        int a = i; b = j;
        if (i > n) a = 2*n - i;
        if (j > n) b = 2*n - j;
        System.out.print(n + 1 - Math.min(a, b) + " ");
    }
    System.out.println();
}
}
```

Explanation

→ n ⇒ 4

1st try to print this using min(i,j)

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

k

min(i,j)

Printing min(i,j) bcz both indexes you will get this.

3 2 1 → b

1 2 3 4 5 6 7 → j

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

min(a,b)

a i

Now we need to convert it using

min(a, b).

To convert we use this.

⇒ k + min(a, b) = n + 1

⇒ k = n + 1 - min(a, b).

a + i = 2n ; b + j = 2n.

b ⇒ 2n - i ; b ⇒ 2n - j.

Now when (i > n) & (j > n). we need to change indexes from "5, 6, 7" to "3, 2, 1". to do this we use this.