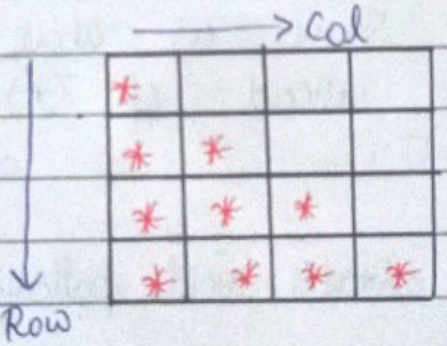
• Pattern printing :-

Pattern programs are design consisting of numbers, alphabets in a perticular form.

→ Trick :-

i) Run the outer for loop for row (no of line)
ii) Identify for every row no how many col are there on types of element in col.
iii) what do you need to print.

① Pattern 1 :-



logic ⇒ No of row = No of column.

Function ⇒

```
Static void pattern1 (int n)
{
    for (int i=1; i<=n; i++)
    {
```

Date: / / Page no:

```
    for (int j=0; j<=i; j++)
    {
        Sout ("* ");
    }
        Sout ();
    }
}
```

② Pattern 2 :-

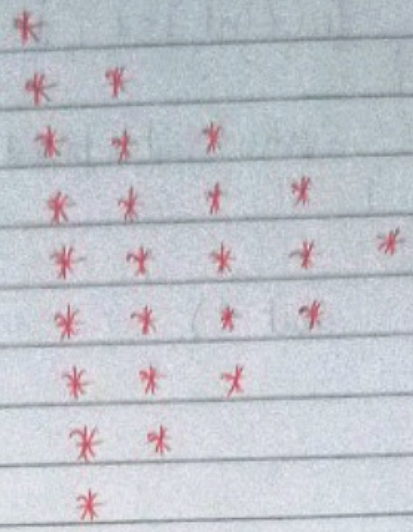
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

logic :- Same as above just print j instead of (*)

fuction :-

```
Static void pattern2 (int n)
{
    for (int i=1; i<=n; i++)
    {
        for (int j=1; j<=i; j++)
        {
            Sout (j+" ");
        }
            Sout ();
    }
}
```

- **Pattern 3 :-**



Logic ⇒ No of rows = (2n-1) and
if (row <= n) :
print same no of column.
else :
col = 2n-row

function :-

```
Static void pattern3 (int n)
{
for (int i=1 ; i<= 2n-1 ; i++)
{
if (i<=n) {
Sout
for (int j=1 ; j<=i ; j++)
{
Sout ("* ");
}
}
}
```

```
else {
    for (int j=1 ; j<= 2n-i ; j++)
{
    System.out.print ("* ");
    }
}
    sout ();
}
}
```

- Pattern 4 :-

```
        1
      2 1 2
    3 2 1 2 3
  4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

logic :- at each row -
value of col = i to 1 and 2 to i

fuction :-

```
Static void pattern4 (int n)
{
  for (int i=0 ; i<n ; i++)
{
  for (int j = 0 ; j<n-i ; j++)
{
    Sout (" ");
  }
  for (int j=i ; j>0 ; j--)
```

Date: / / Page no: 43

```
{
    Sout (j+" ") ;
}
for (int j=2 ; j<=i ; j++)
{
    Sout (j+" ") ;
}
Sout () ;
}
}
```