# CSE1121
# Structured & OOP Language

## *Pattern Printing*

Sumaya Kazary

Assistant Professor

Department of Computer Science and Engineering

Dhaka University of Engineering & Technology, Gazipur

**Acknowledgement**

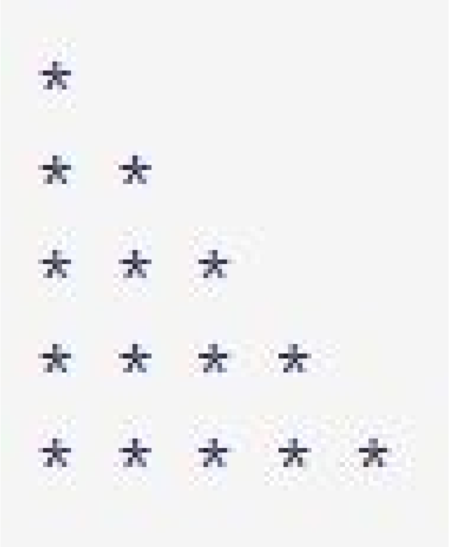
Thanks to the authors of all the books and online tutorials used in this slide.

# Pattern Printing in C Language

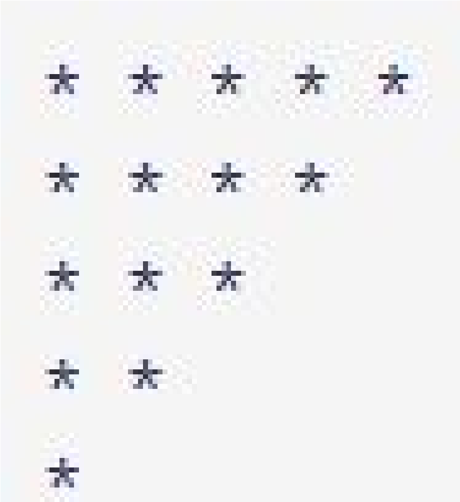
- What is Pattern in C Language
- Different Types of Pattern in C Language
- Preliminaries for Pattern Printing: Nested for loop
- Secret of Pattern Printing in C Language
- Discover the Secret of Pattern Printing
- How Print Patterns with Loops in C
- Pattern Printing Example With Real Program.

# What is Pattern in C language?

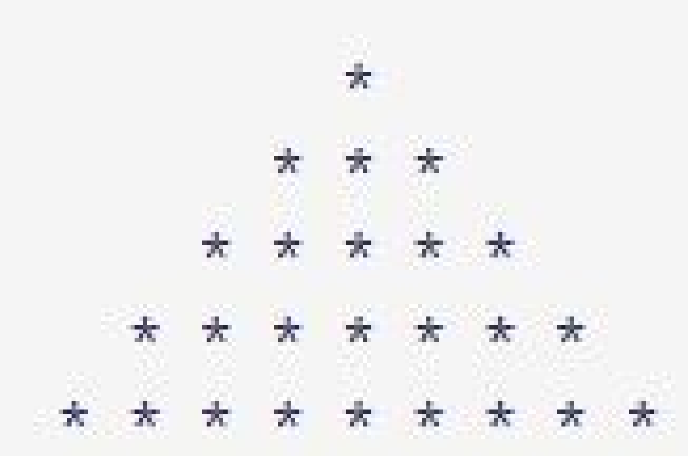
- When we display something with some special order then it’s called a pattern.



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

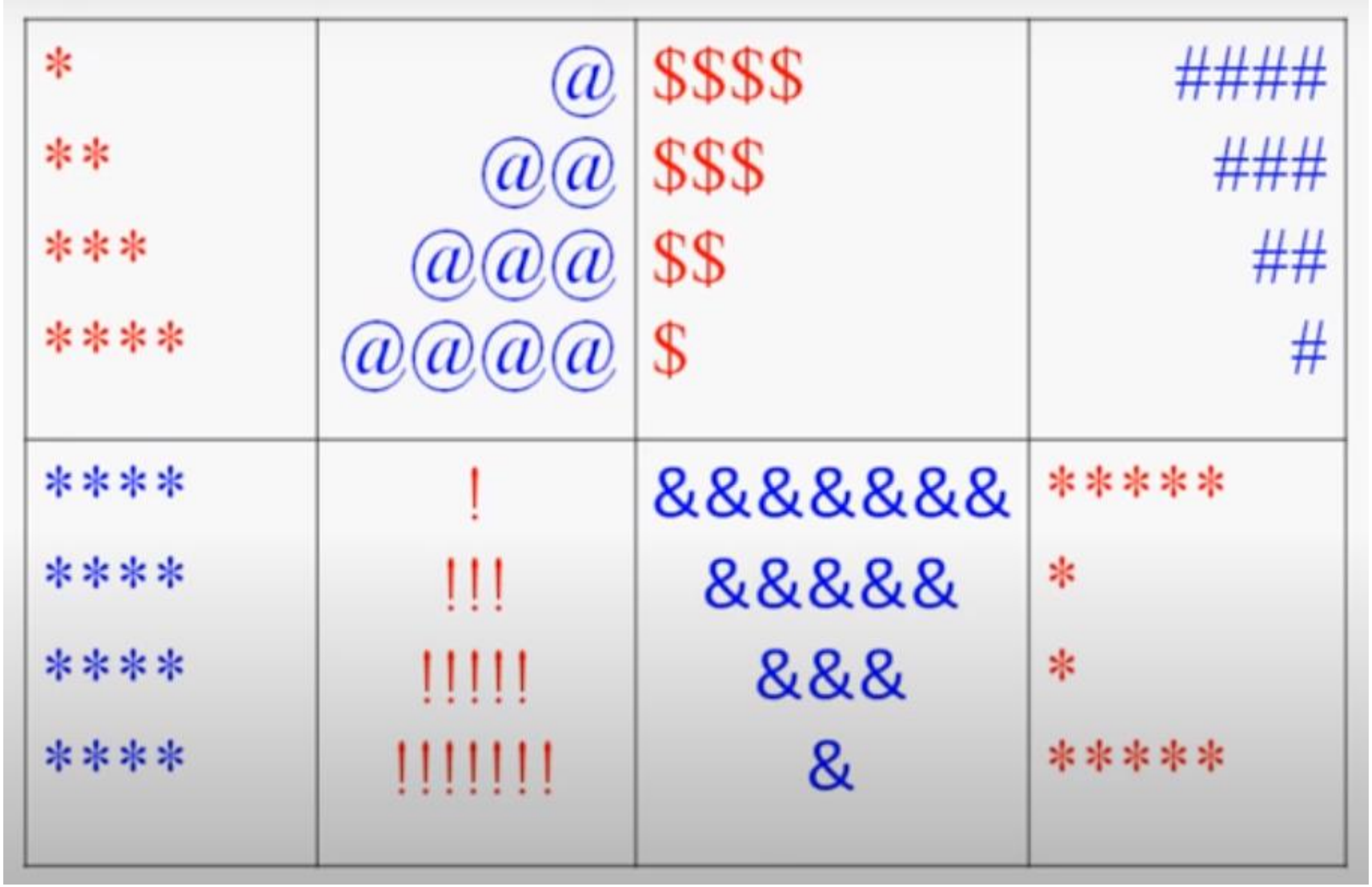


```
* * * * *
* * * *
* * *
* *
*
```



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

# Different types of pattern in C Language



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

```
   @
  @@
 @@@
@@@@
```

```
$$$$
$$$
$$
$
```

```
####
 ###
  ##
   #
```

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

```
   !
  !!!
 !!!!!
!!!!!!!
```

```
&&&&&&&
 &&&&&
  &&&
   &
```

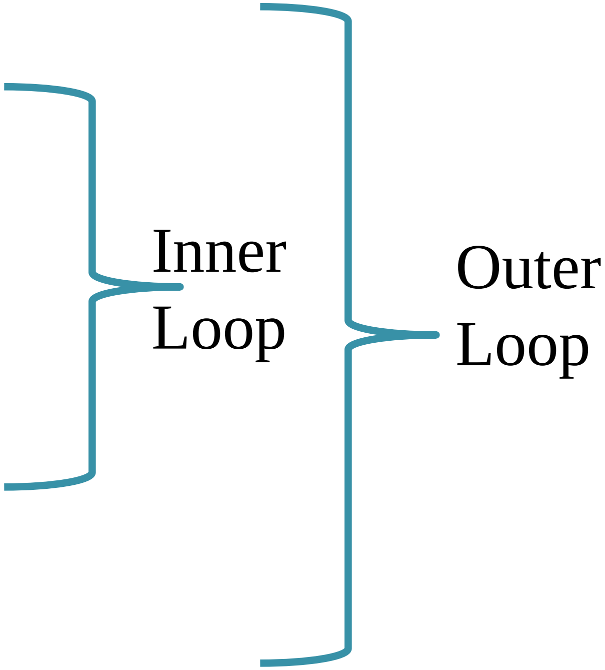
```
*****
*
*
*****
```

# Preliminaries for Pattern Printing:

## Nested *for* loop

- The syntax for a **nested for loop** statement in C is as follows −

```
for ( init;  condition;  increment )
{  for ( init; condition; increment )
        {
         // statement of inside loop;
          }
     // statement of outer loop
 }
```



Inner Loop

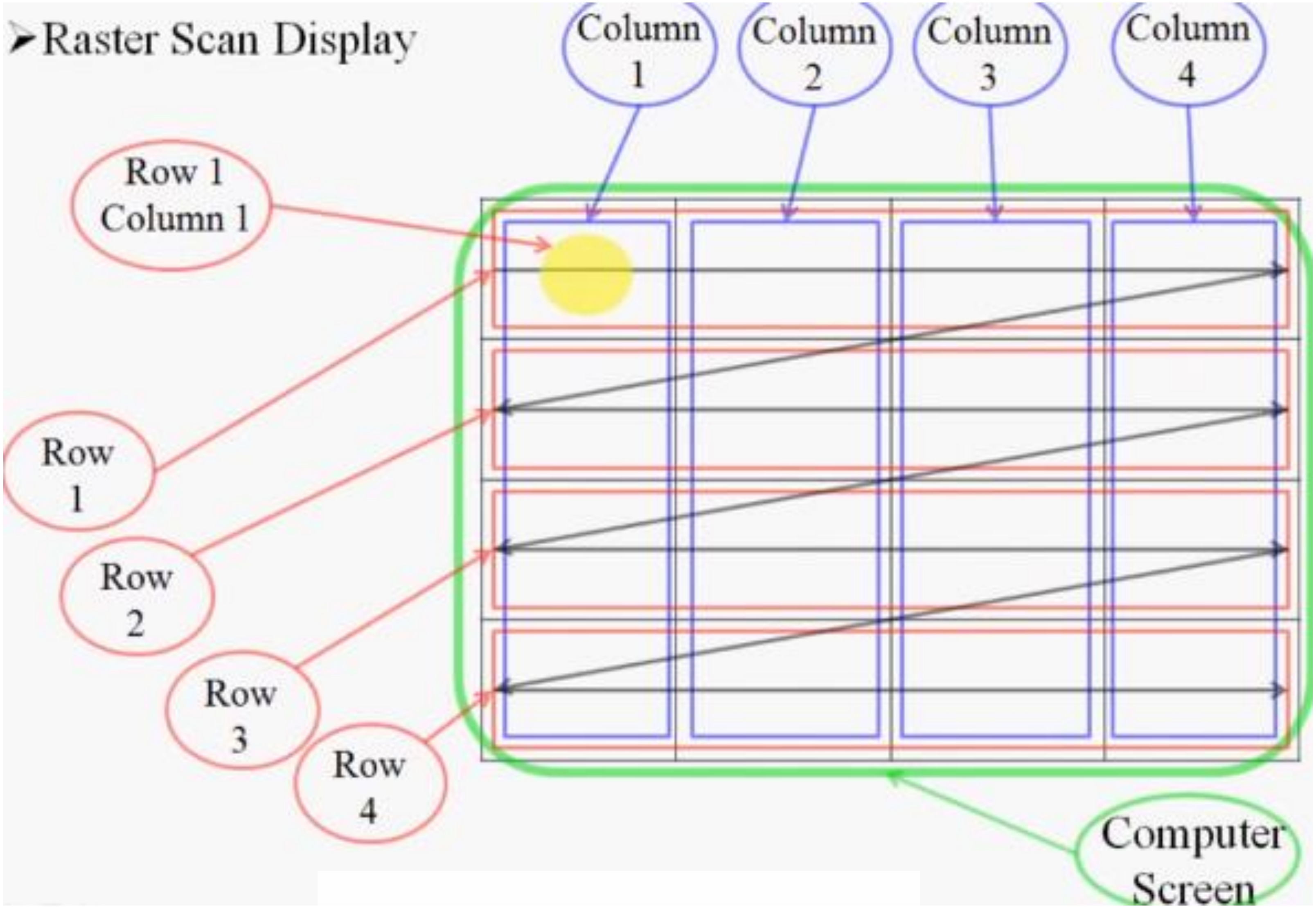
Outer Loop

# Nested for loop Example:

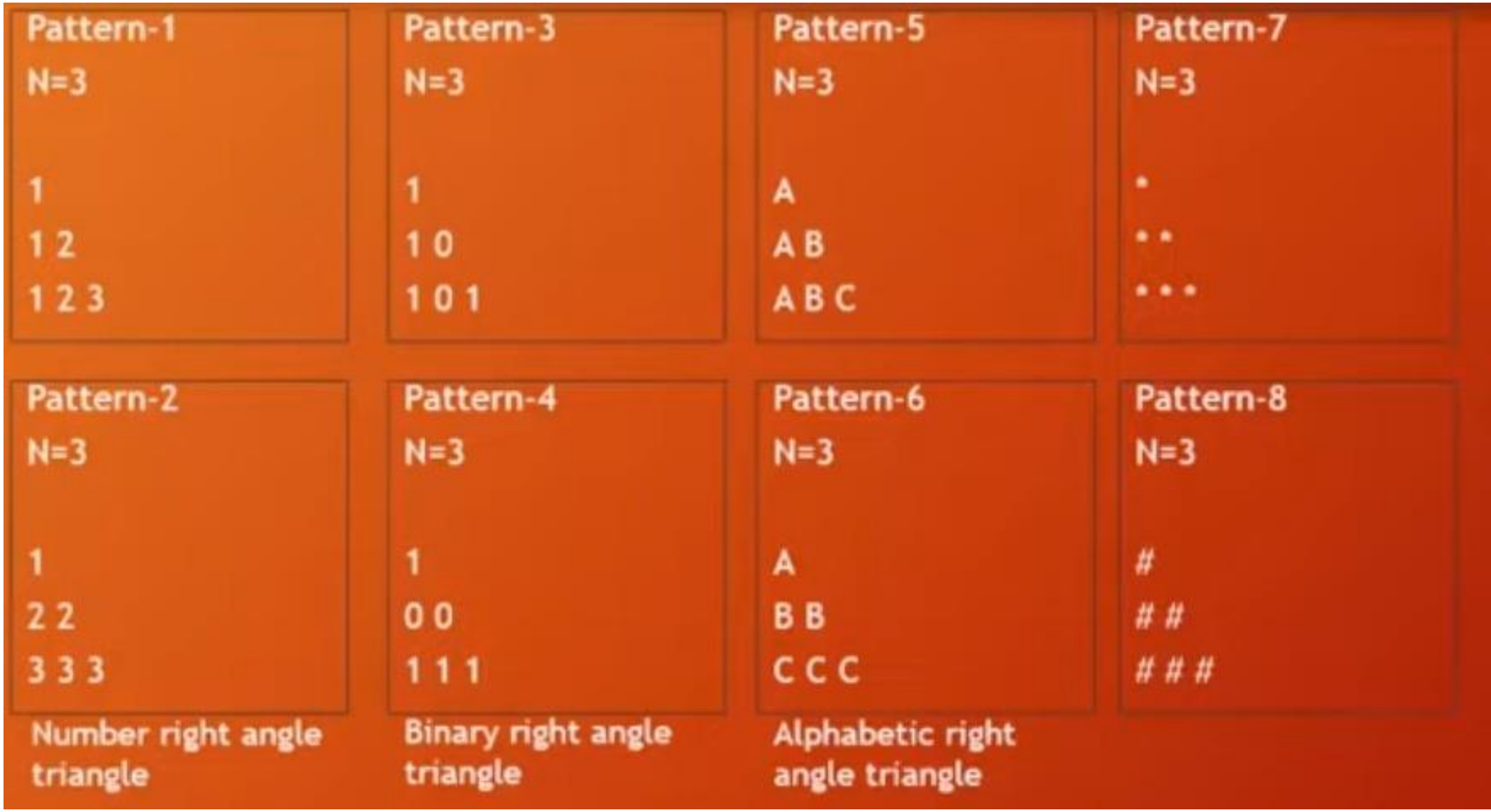
## Multiplication Table from 1 to 5

```
int main( ) {
 int num, i;
 for(num=1;num<=5;num++)
 {  printf("Multiplication Table of %d\n",num);
     for (i = 1; i <= 10; i++)
      printf("%d x %d= %d\n", num, i, num * i);
  }
 return 0;
}
```

# Secret of pattern printing in C Language



# Pattern Category I



Pattern-1

N=3

```
1
1 2
1 2 3
```

Pattern-3

N=3

```
1
1 0
1 0 1
```

Pattern-5

N=3

```
A
A B
A B C
```

Pattern-7

N=3

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

Pattern-2

N=3

```
1
2 2
3 3 3
```

Number right angle triangle

Pattern-4

N=3

```
1
0 0
1 1 1
```

Binary right angle triangle

Pattern-6

N=3

```
A
B B
C C C
```

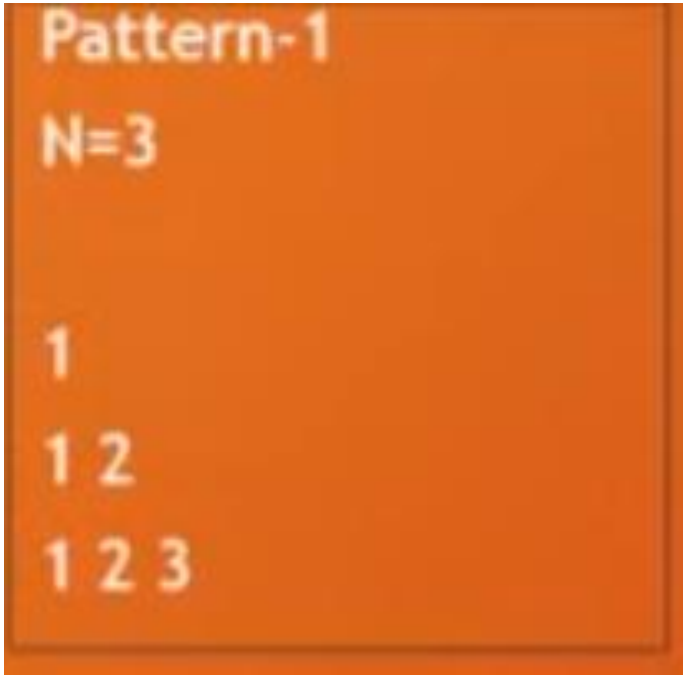
Alphabetic right angle triangle

Pattern-8

N=3

```
#
# #
# # #
```

# Lets start with Pattern 1



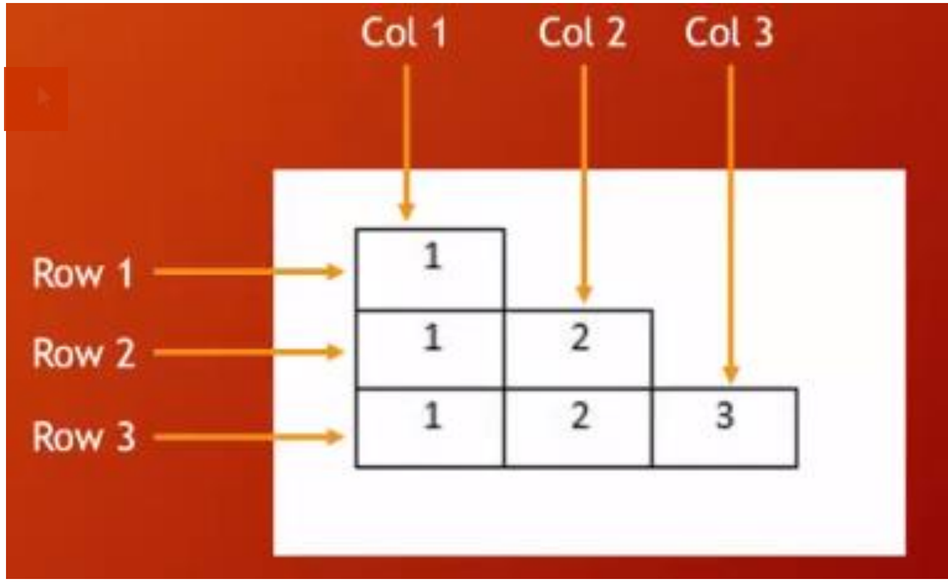
Pattern-1

N=3

1

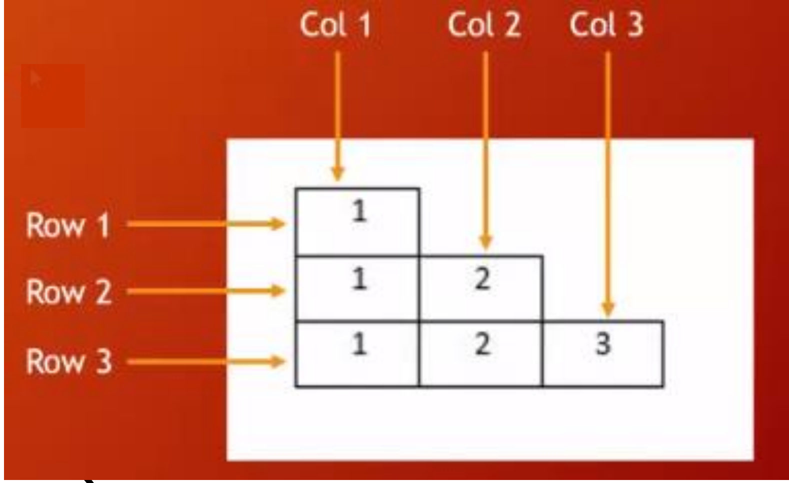
1 2

1 2 3



|  | Col 1 | Col 2 | Col 3 |
|---|---|---|---|
| Row 1 | 1 |  |  |
| Row 2 | 1 | 2 |  |
| Row 3 | 1 | 2 | 3 |

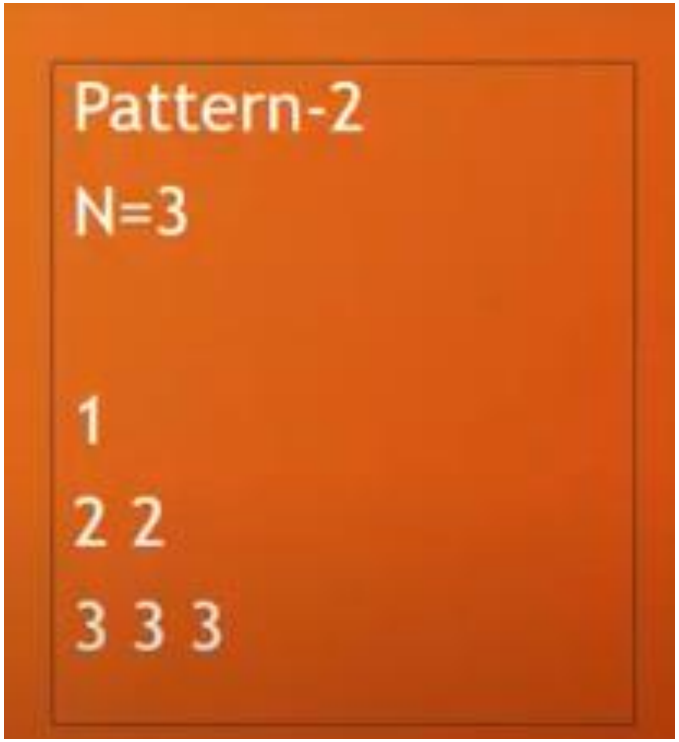
# Pattern 1 Loop:



| | Col 1 | Col 2 | Col 3 |
|---|---|---|---|
| Row 1 | 1 | | |
| Row 2 | 1 | 2 | |
| Row 3 | 1 | 2 | 3 |

```
for(row=1;row<=n; row++)
    {
      for(col=1; col<=row; col++)
      {
           printf("%d ", col);
      }
     printf("\n");
    }
```

# Lets start with Pattern 2



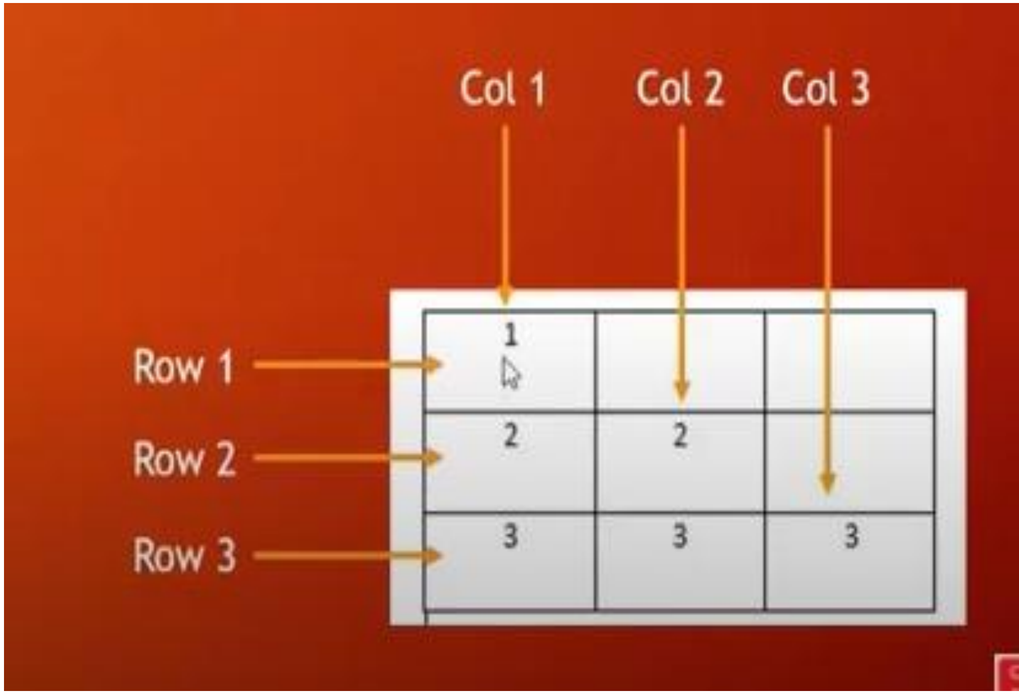
Pattern-2

N=3

1

2 2

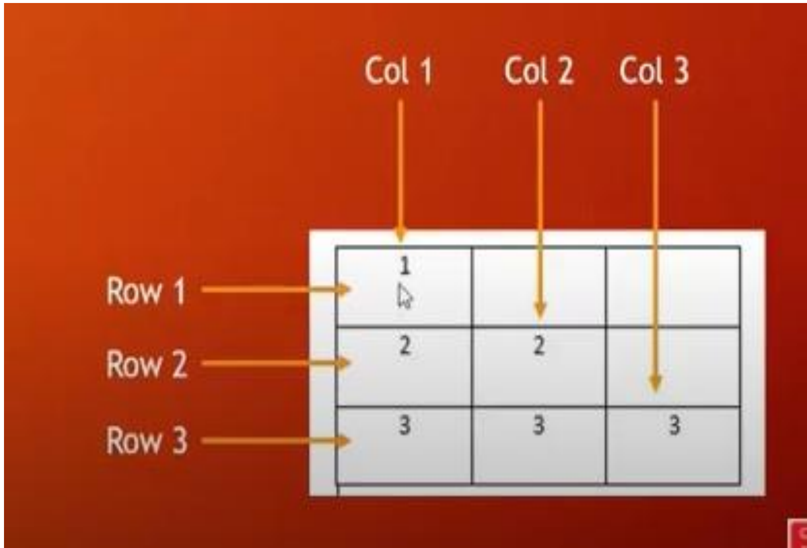
3 3 3



| | Col 1 | Col 2 | Col 3 |
|---|---|---|---|
| Row 1 | 1 | | |
| Row 2 | 2 | 2 | |
| Row 3 | 3 | 3 | 3 |

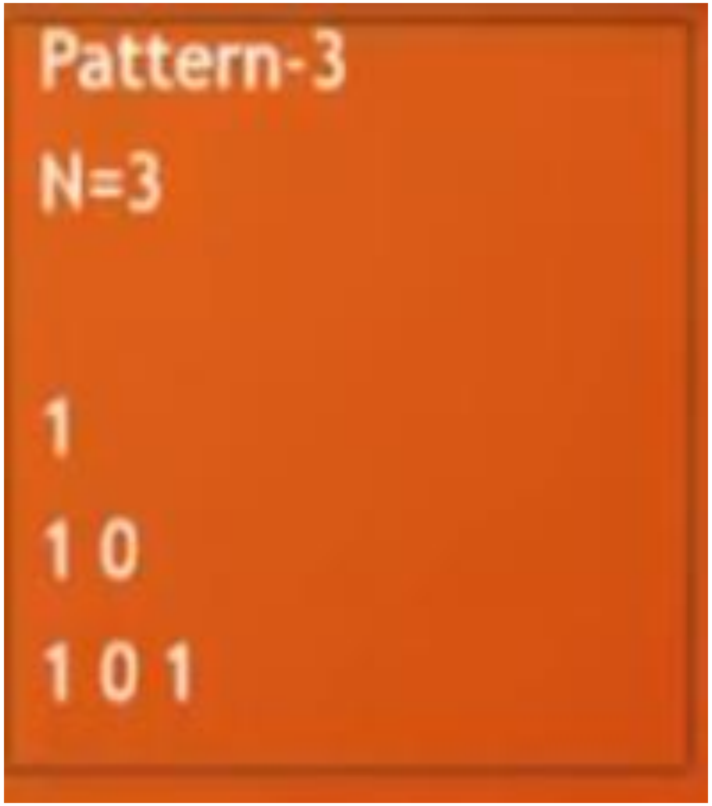
# Pattern 2 Loop:

```
for(row=1;row<=n;row++)
    {
      for(col=1; col<=row; col++)
      {
           printf("%d ", row);
      }
     printf("\n");
    }
```



|       | Col 1 | Col 2 | Col 3 |
|-------|-------|-------|-------|
| Row 1 | 1     |       |       |
| Row 2 | 2     | 2     |       |
| Row 3 | 3     | 3     | 3     |

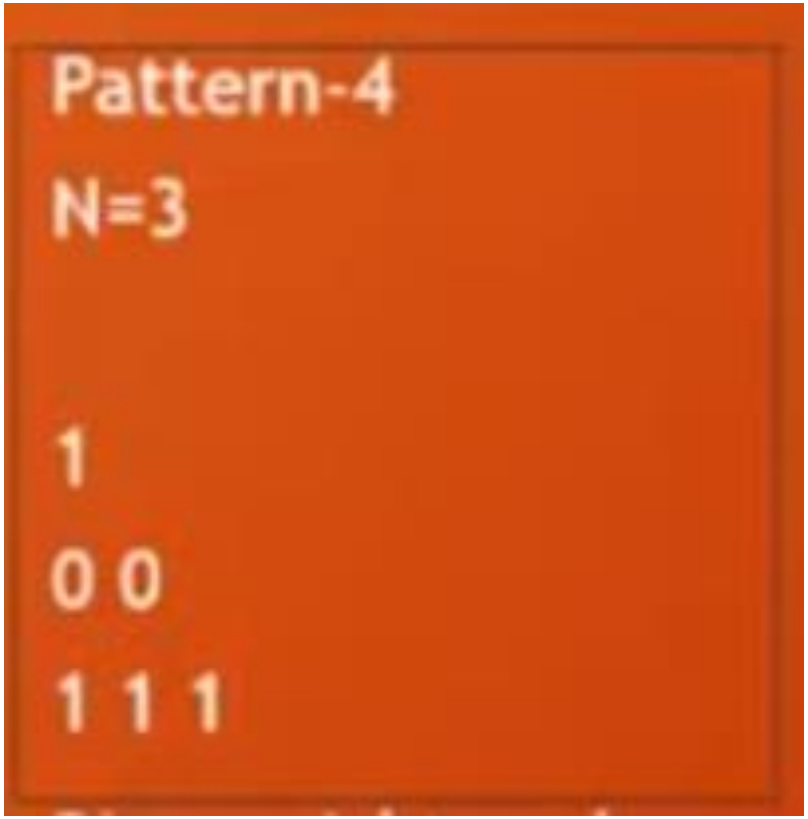
# Lets start with Pattern 3 & 4



Pattern-3

N=3

```
1
1 0
1 0 1
```

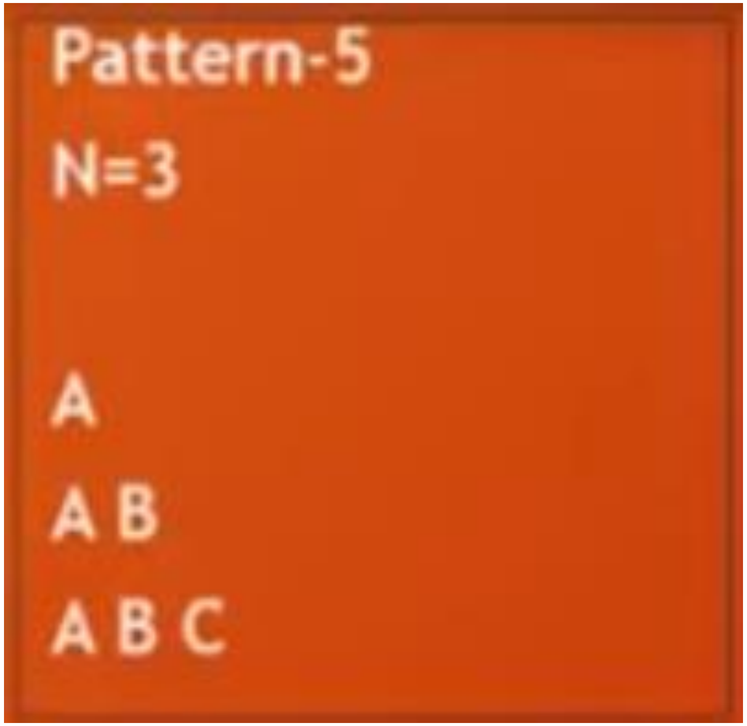


Pattern-4

N=3

```
1
0 0
1 1 1
```

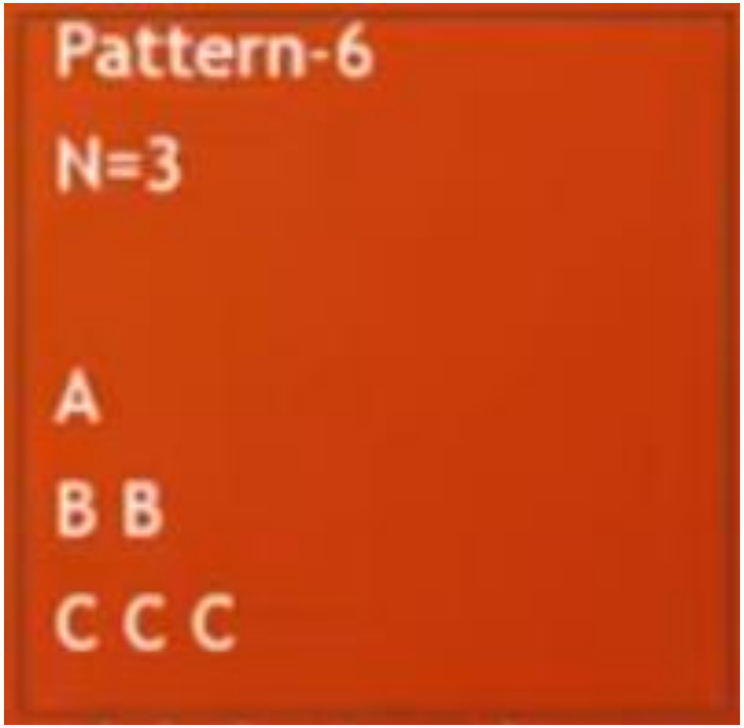
# Lets start with Pattern 5 & 6



Pattern-5

N=3

```
A
A B
A B C
```

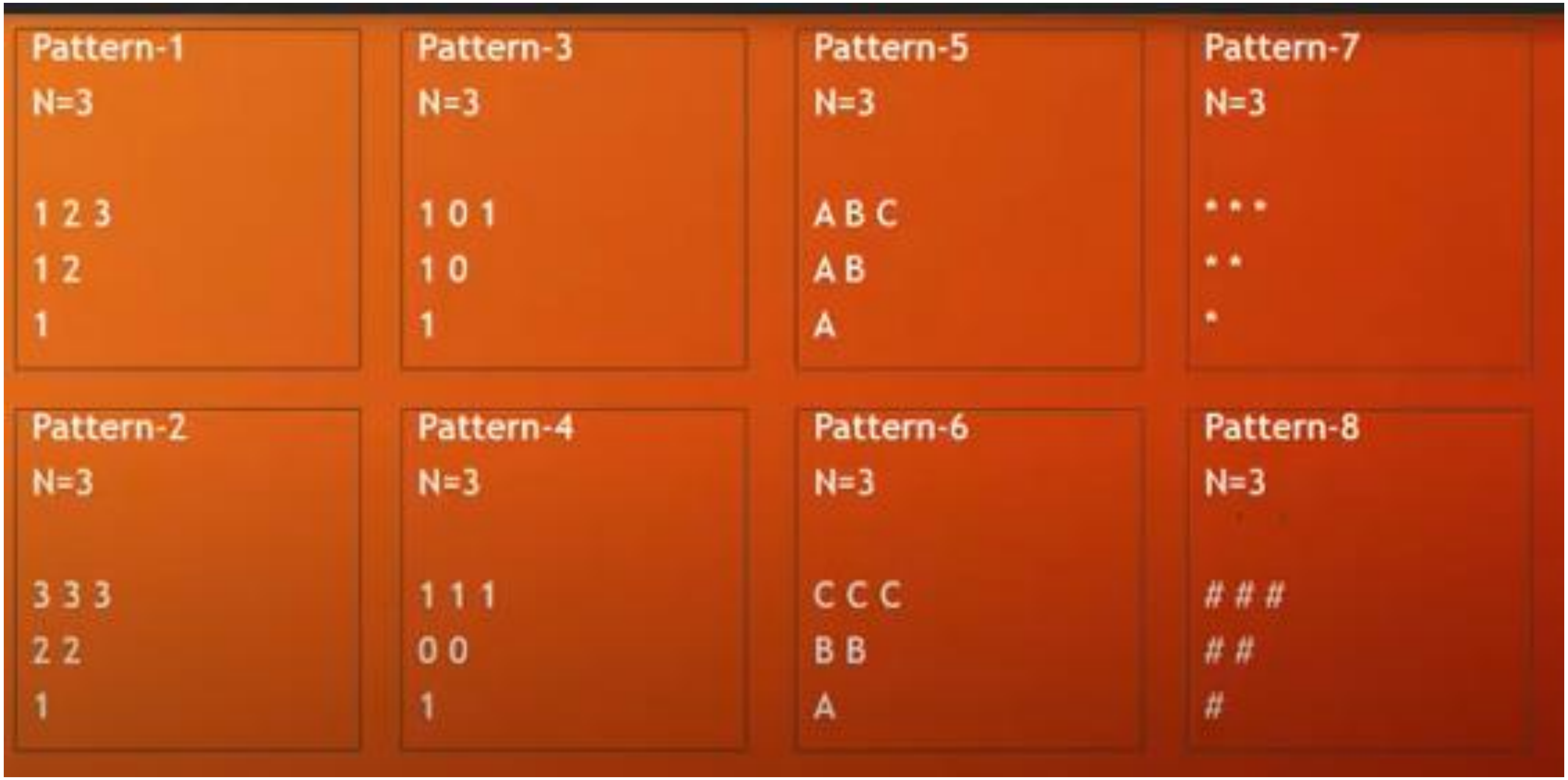


Pattern-6

N=3

```
A
B B
C C C
```

# Pattern Category II



Pattern-1

N=3

```
1 2 3
1 2
1
```

Pattern-2

N=3

```
3 3 3
2 2
1
```

Pattern-3

N=3

```
1 0 1
1 0
1
```

Pattern-4

N=3

```
1 1 1
0 0
1
```

Pattern-5

N=3

```
A B C
A B
A
```

Pattern-6

N=3

```
C C C
B B
A
```

Pattern-7

N=3

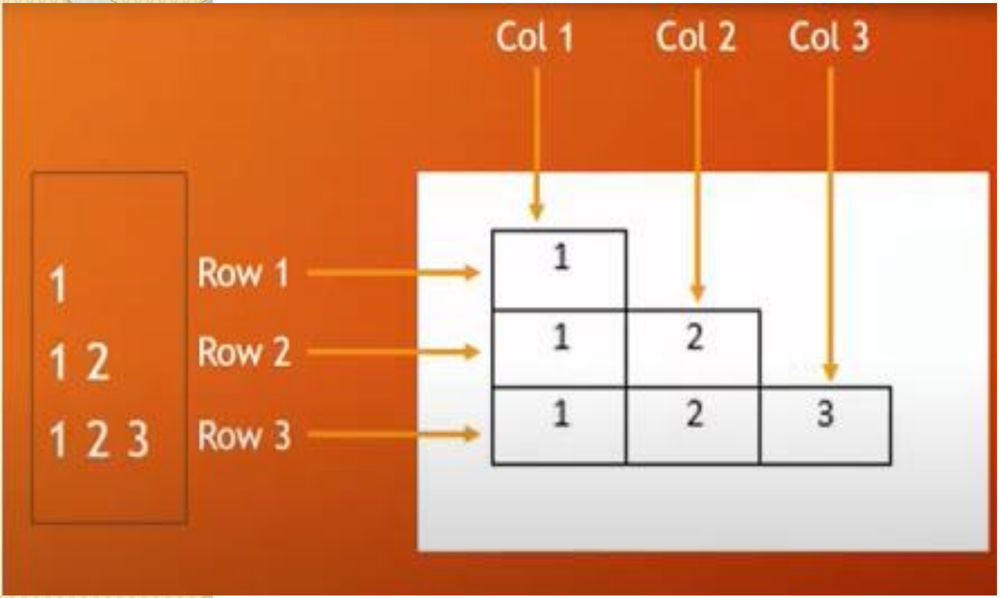
```
* * *
* *
*
```

Pattern-8

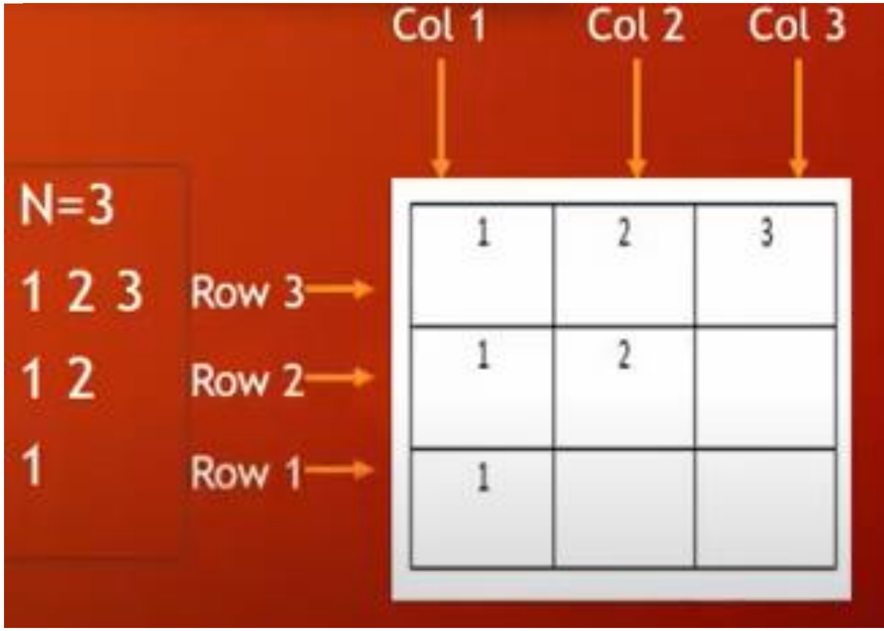
N=3

```
# # #
# #
#
```

# Lets start with Pattern II(1)



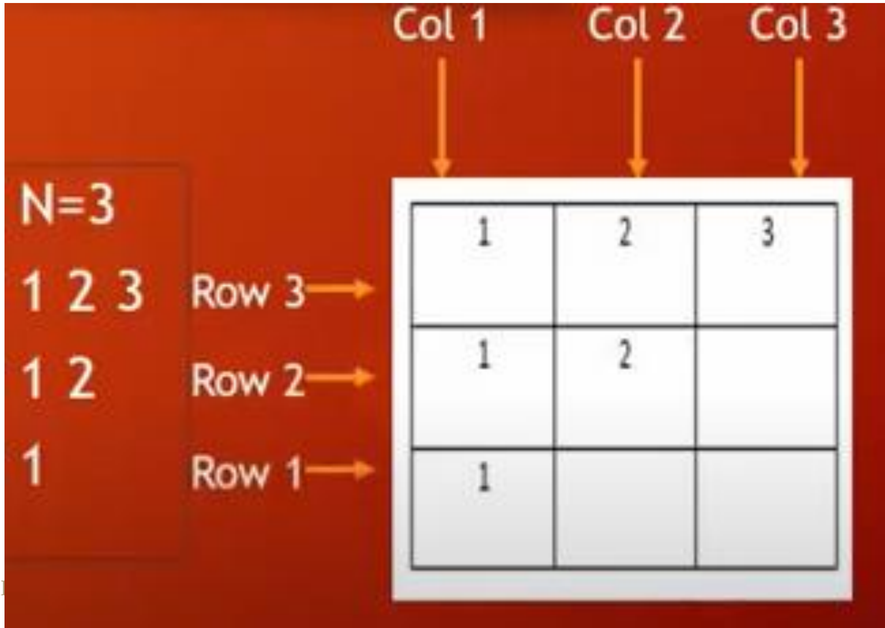
1  
1 2  
1 2 3



N=3  
1 2 3  
1 2  
1

# Pattern II(1) Loop:

```
for(row=n;  row>=1;  row--)
 {
   for(col=1;  col<=row;  col++)
   {
         printf("%d ", col);
   }
  printf("\n");
 }
```

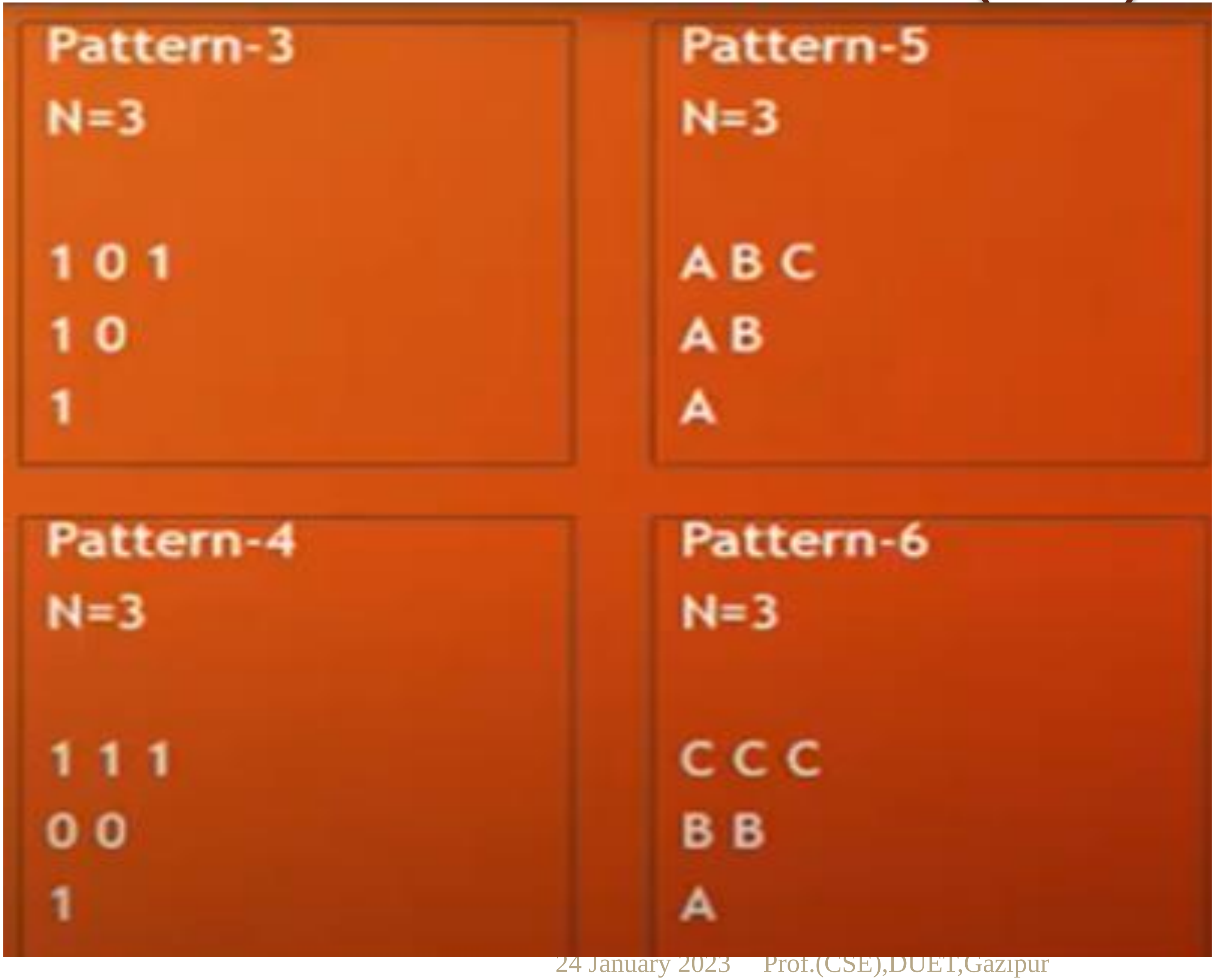


N=3

```
1 2 3
1 2
1
```

|  | Col 1 | Col 2 | Col 3 |
|---|---|---|---|
| Row 3 | 1 | 2 | 3 |
| Row 2 | 1 | 2 |  |
| Row 1 | 1 |  |  |

# Lets start with Pattern II(3-6)



Pattern-3

N=3

```
1 0 1
1 0
1
```

Pattern-5

N=3

```
A B C
A B
A
```

Pattern-4

N=3

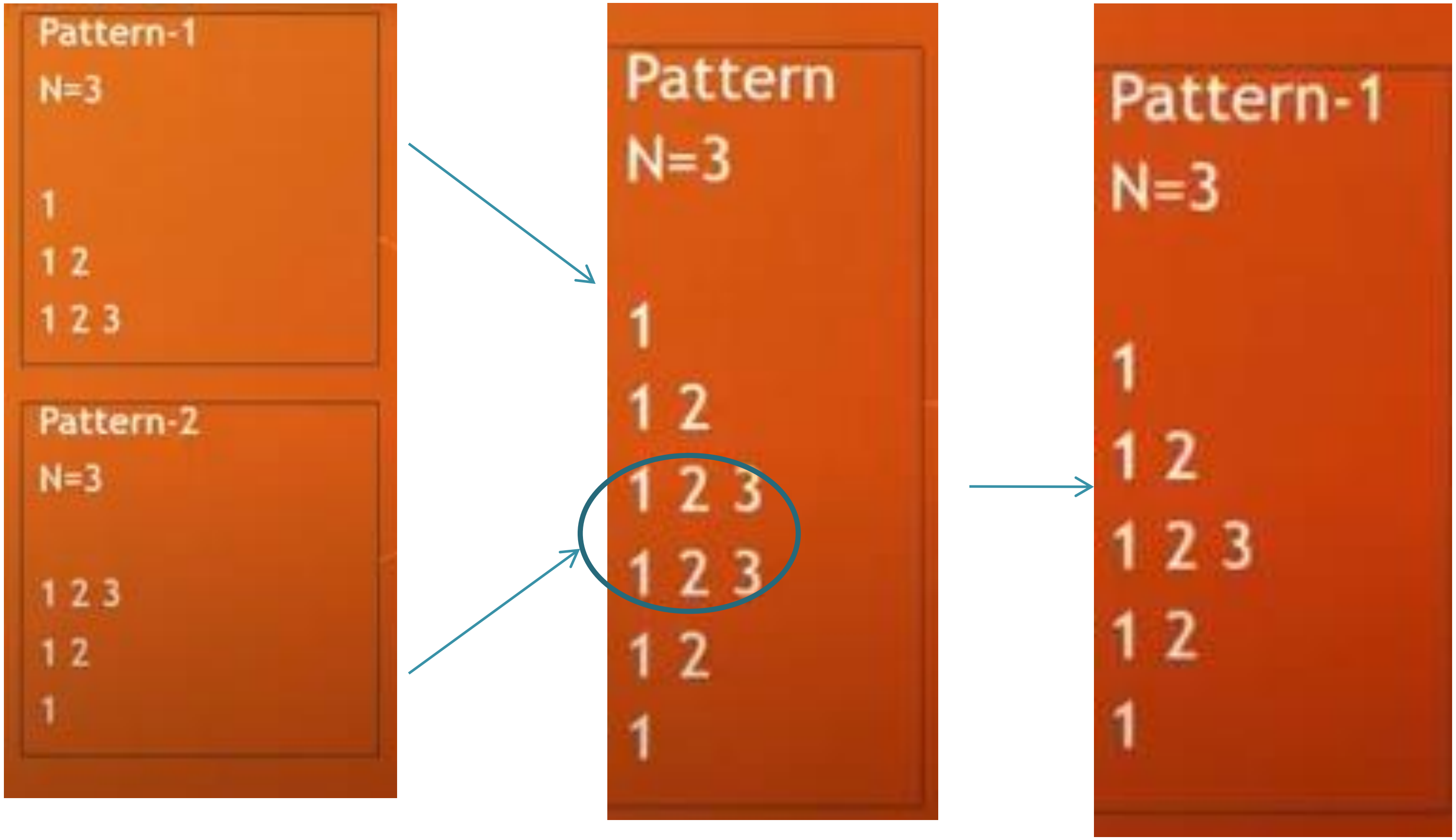
```
1 1 1
0 0
1
```

Pattern-6

N=3

```
C C C
B B
A
```

# Home task :Pattern Category III



```
Pattern-1
N=3

1
1 2
1 2 3
```

```
Pattern-2
N=3

1 2 3
1 2
1
```

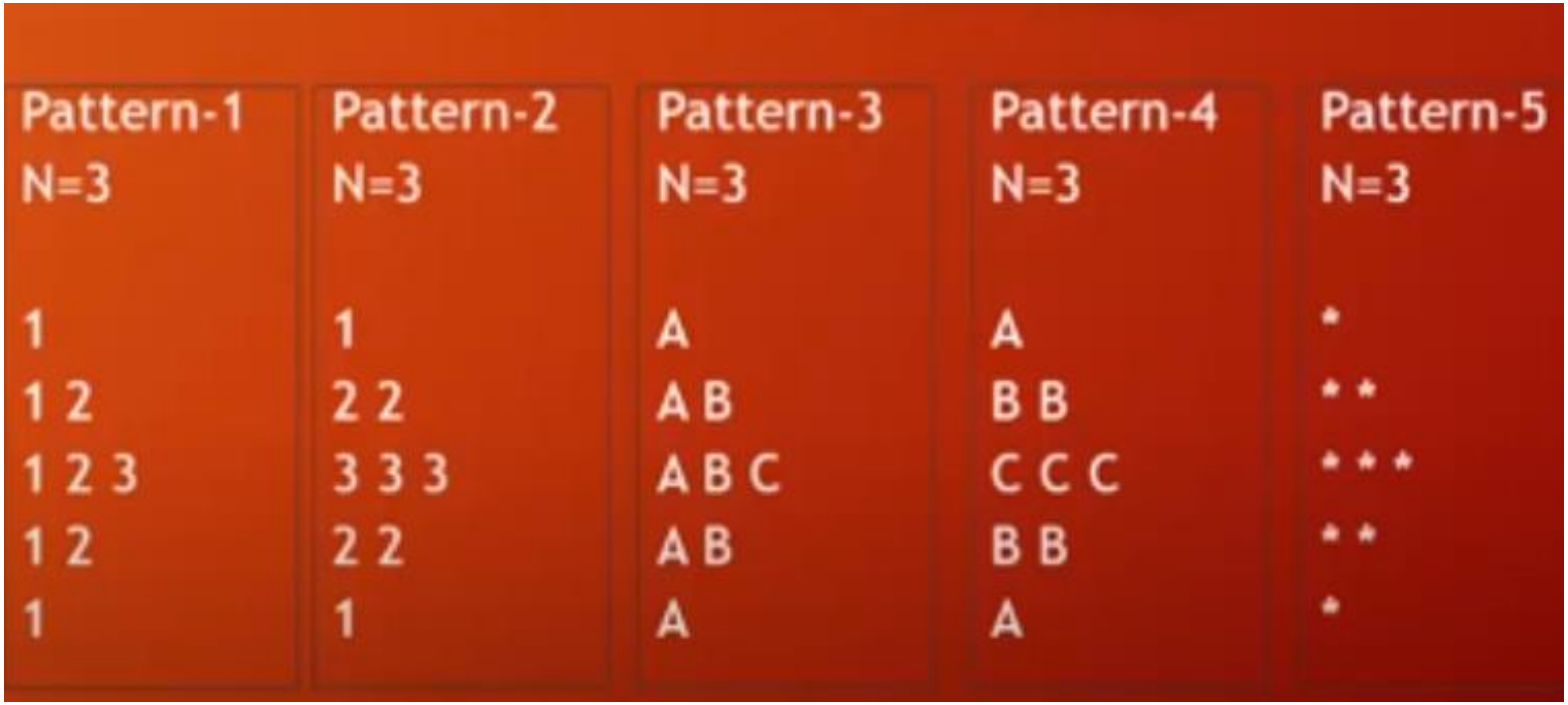
```
Pattern
N=3

1
1 2
1 2 3
1 2 3
1 2
1
```

```
Pattern-1
N=3

1
1 2
1 2 3
1 2
1
```

# Home task :Pattern Category III



| Pattern-1 | Pattern-2 | Pattern-3 | Pattern-4 | Pattern-5 |
|---|---|---|---|---|
| N=3 | N=3 | N=3 | N=3 | N=3 |
| 1 | 1 | A | A | * |
| 1 2 | 2 2 | A B | B B | * * |
| 1 2 3 | 3 3 3 | A B C | C C C | * * * |
| 1 2 | 2 2 | A B | B B | * * |
| 1 | 1 | A | A | * |

# Thank You