Nested for loop (pattern program)

6.1)

	c=1	2	3	4	5	6
r=1	*					
r=2	*	*				
r=3	*	*	*			
r=4	*	*	*	*		
r=5	*	*	*	*	*	
r=6	*	*	*	*	*	*

1. row is responsible for changing the line
2. column is responsible for printing the value
3. Inner for loop is column and outer for loop is row
4. row always depends on no of lines
5. column always depends on value of row

```
for (r=1 ; r<=n ; r++)
{
    for (c=1 ; c<=r ; c++)
    {
        printf("* ");
    }
    printf("\n");
}
```

Dry Run

n=6

r=1 c=1 2

r=2 c=1 2 3

r=3 c=1 2 3 4

r=4 c=1 2 3 4 5

r=5 c=1 2 3 4 5 6

r=6 c=1 2 3 4 5 6 7

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

Nested for loop (pattern program)

	C=1	2	3	4	5
r=5	*	*	*	*	*
r=4	*	*	*	*	
r=3	*	*	*		
r=2	*	*			
r=1	*				

1) row is responsible for changing the line
2) column is responsible for printing the value
3) Inner for loop is column and outer for loop is row
4) row always depends on no of lines
5) column always depends on value of row

```
for (r=n ; r>=1; r--)
{
    for (c=1 ; c<=r; c++)
    {
        printf("* ");
    }
    printf("\n");
}
```

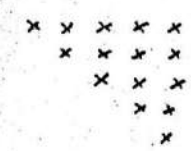
Dry Run

n = 5

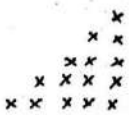
r=5 c=1 2 3 4 5
r=4 c=1 2 3 4
r=3 c=1 2 3
r=2 c=1 2
r=1 c=1

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

(1)



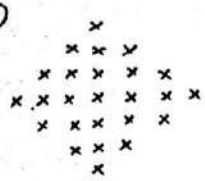
(2)



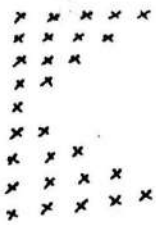
(3)

(4)

(5)



(6)



(7)

(8)

(9)

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

(10)

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

(11)

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

(12)

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

(13)

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

(14)

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

(15)

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

(16)

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

(17)

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

(pascal triangle)

(18)

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

(19)

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

(Floyed Triangle)

(20)

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

(21)

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

(22)

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

(23)

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

(24)