31/08/2023
Thursday

1. **PATTERN 1 →**

```
- - - - *  → R0
- - - * *  → R1
- - * * *  → R2
- * * * *  → R3
* * * * *  → R4
↓ ↓ ↓ ↓ ↓
C0 C1 C2 C3 C4
```

Total rows = (n = 5)

R0 → 4 space, 1 *
R1 → 3 space, 2 *
R2 → 2 space, 3 *
R3 → 1 space, 4 *
R4 → 0 space, 5 *

Conditions –

// spaces →
col < n - row - 1
Print "_"

// stars →
col < row + 1
Print "*"

2. **PATTERN 2 →**

```
R0 → - - - *
R1 → - - * _ *
R2 → - * _ * _ *
R3 → * _ * _ * _ *
```

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

Total rows = (n = 4)

Conditions –

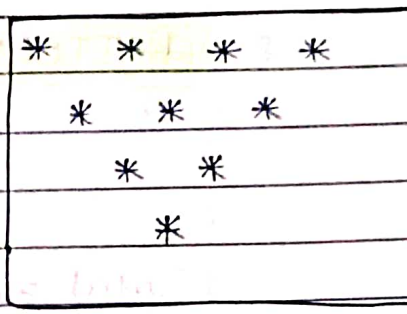
R0 → 3 space, 1 star
R1 → 2 space, 2 star
R2 → 1 space, 3 star
R3 → 0 space, 4 star

// spaces →
col < n - row - 1
Print "_"

// stars →
col < row + 1
Print "* _"

__/__/____

3. **PATTERN 3 →**



R0 → * _ * _ * _ *
R1 → _ * _ * _ *
R2 → _ _ * _ *
R3 → _ _ _ *

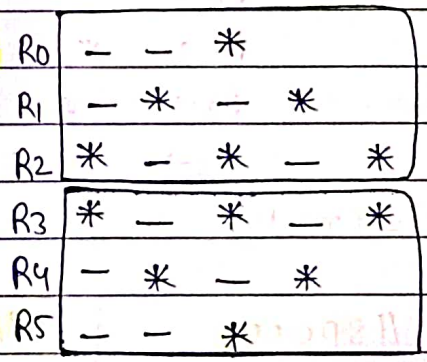
Total rows = (n=4)

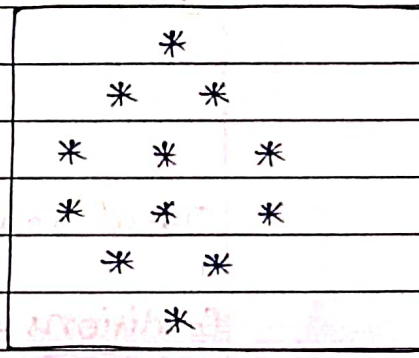
R0 → 0space, 4star
R1 → 1space, 3star
R2 → 2 space, 2star
R3 → 3 space, 1star

Conditions –

// spaces
col < row
Print "_"
// stars
col < n-row
Print "*_"

4. **PATTERN 4 →**





R0 — — *
R1 — * — *
R2 * — * — *
R3 * — * — *
R4 — * — *
R5 — — *

Total rows = (n=6)

R0 → 2space, 1star
R1 → 1space, 2star
R2 → 0space, 3star
R3 → 0 space, 3star
R4 → 1 space, 2star
R5 → 2 space, 1star

Conditions for upper triangle →

// spaces
col < n-row-1, print "_"
// stars
col < row+1, print "*_"

Condition for lower triangle →

// spaces
col < row, print "_"
// stars
col < n-row, print "*_"

__/__/____

5. **PATTERN 5 —**

R0→ *
R1→ * *
R2→ * *
R3→ * *

Total rows = (n = 4)

Conditions —	// Spaces	// Stars

	col < n - row - 1
	print "_"
		col < row + 1
		if (col == 0 || col = row + 1 - 1)
		print "*_"
		else
		print "__"

6. **PATTERN 6 —**

R0→ * *
R1→ * *
R2→ * *
R3→ *

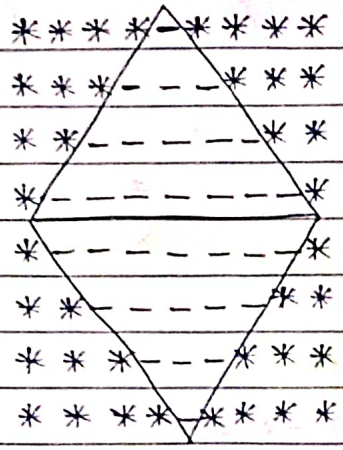
Total rows = (n = 4)

Conditions -	// Spaces	// stars

	col < row
	print "_"
		col < n - row
		if (col == 0 || col == n - row - 1)
		print "*_"
		else
		print "__"

__/__/__

7. PATTERN 7 –



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

CONDITIONS

→ // upper pattern → // Inverted pyramid1

```
col < n - row
print "*"
```

// Triangle

```
col < 2*row + 1
print "_"
```

// Inverted pyramid 2

```
col < n - row
print "*"
```

→ // lower pattern → ~~// Invert~~ // Pyramid 1

```
col < row + 1
print "*"
```

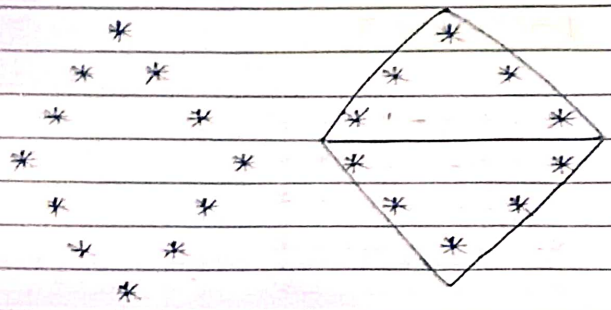
// Inverted triangle

```
col < 2*n - 2*row - 1
print "_"
```

// Pyramid2

```
col < row + 1
print "*"
```

8. PATTERNS –



→ // upper pattern → // spaces

```
col < n-row-1
print " "
```

// stars

```
col < row+1
print "* "
```

→ // lower pattern → // spaces

```
col < row
print " "
```

// stars

```
col < n-row
print "* "
```

9. **PATTERN 9 –**

R0 → 1
R1 → 2 * 2
R2 → 3 * 3 * 3
R3 → 4 * 4 * 4 * 4
C0 C1 C2 C3 C4 C5 C6

R0 → 1 char
R1 → 3 char
R2 → 5 char
R3 → 7 char
So, col < 2*row+1

Conditions – // inner loop

```
col < 2* row +1
if ( col%2 == 0)
  print (row+1)
else
  print " * "
```

i.e
col == odd
print " * "
col == even
print (row no.+1)

10. **PATTERN 10 –**

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

Conditions – // inner loop

```
col < n
if (row==0 || col==0 || col==n-row)
  print " * "
else
  print " _ "
```