Pattern Printing With Python

Problems:

Pattern Name	Explanation	Example Output

1. Right-angled Triangle	A right-angled triangle of stars with `n` rows. Each row contains stars equal to the row number.	`*`
		`**`
		`***`
		`****`
		`*****`

2. Inverted Triangle	An inverted right-angled triangle of stars with `n` rows. Starts with `n` stars, reducing by 1 each row.	`*****`
		`****`
		`***`
		`**`
		`*`

3. Pyramid	A pyramid of stars centered with `n` rows. Stars increase by 2 in each row.	`  *`
		` ***`
		`*****`
		`*******`
		`*********`

4. Inverted Pyramid	A pyramid flipped upside down with `n` rows. Stars decrease by 2 in each row.	`*********`
		` *******`
		`  *****`
		`   ***`
		`    *`

5. Diamond Pattern	A diamond shape with `2n-1` rows. Combines a pyramid and inverted pyramid.	`    *`
		`   ***`
		`  *****`
		` *******`
		`*********`
		` *******`
		`  *****`
		`   ***`
		`    *`

6. Number Triangle	A right-angled triangle of increasing numbers, starting from 1 in each row.	1 12 123 1234 12345
7. Reversed Number Triangle	A right-angled triangle with numbers in reverse order, starting from the row number.	1 21 321 4321 54321
8. Palindromic Pyramid	A pyramid with numbers that form a palindromic sequence in each row.	1 121 12321 1234321 123454321
9. Alphabet Triangle	A triangle with alphabets increasing row by row.	A AB ABC ABCD ABCDE
10. Checkerboard Pattern	An alternating star (*) and space pattern forming a grid of n x n.	* * * * * * * * * * * * * * * * * * * * * * *
11. Hollow Square	A square with * border and empty spaces inside for n x n.	***** *****
12. Floyd's Triangle	Numbers arranged in rows, starting from 1 and increasing sequentially.	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

13. Butterfly Pattern	A mirrored triangle on top and bottom, forming a butterfly shape.	********
14. Pascal's Triangle	A triangular arrangement of binomial coefficients for `n` rows.	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
15. Zigzag Pattern	A zigzag pattern with alternating stars (`*`) in `n` rows and specified length.	* * * * * * * * * * *

Solution:

1. Right-angled Triangle

```
n = 5
for i in range(1, n + 1):
    print('*' * i)
```

---

2. Inverted Right-angled Triangle

```
n = 5
for i in range(n, 0, -1):
    print('*' * i)
```

---

3. Pyramid Pattern

```
n = 5
for i in range(1, n + 1):
    print(' ' * (n - i) + '*' * (2 * i - 1))
```

---

4. Inverted Pyramid

```
n = 5
for i in range(n, 0, -1):
    print(' ' * (n - i) + '*' * (2 * i - 1))
```

---

5. Diamond Pattern

```
n = 5
# Upper part
for i in range(1, n + 1):
    print(' ' * (n - i) + '*' * (2 * i - 1))
# Lower part
for i in range(n - 1, 0, -1):
    print(' ' * (n - i) + '*' * (2 * i - 1))
```

---

6. Number Triangle

```
n = 5
for i in range(1, n + 1):
    print(''.join(str(j) for j in range(1, i + 1)))
```

---

7. Reversed Number Triangle

```
n = 5
```

```
for i in range(1, n + 1):
    print(''.join(str(j) for j in range(i, 0, -1)))
```

---

8. Palindromic Pyramid

```
n = 5
for i in range(1, n + 1):
    left = ''.join(str(j) for j in range(1, i + 1))
    right = ''.join(str(j) for j in range(i - 1, 0, -1))
    print(' ' * (n - i) + left + right)
```

---

9. Alphabet Triangle

```
n = 5
for i in range(1, n + 1):
    print(''.join(chr(65 + j) for j in range(i)))
```

---

10. Checkerboard Pattern

```
n = 5
for i in range(n):
    if i % 2 == 0:
        print('* ' * n)
    else:
        print(' *' * n)
```

---

11. Hollow Square

```
n = 5
for i in range(n):
    if i == 0 or i == n - 1:
```

```
        print('*' * n)
    else:
        print('*' + ' ' * (n - 2) + '*')
```

---

12. Floyd's Triangle

```
n = 5
num = 1
for i in range(1, n + 1):
    for j in range(i):
        print(num, end=' ')
        num += 1
    print()
```

---

13. Butterfly Pattern

```
n = 4
# Upper part
for i in range(1, n + 1):
    print('*' * i + ' ' * (2 * (n - i)) + '*' * i)
# Lower part
for i in range(n - 1, 0, -1):
    print('*' * i + ' ' * (2 * (n - i)) + '*' * i)
```

---

14. Pascal's Triangle

```
n = 5
for i in range(n):
    # Print spaces for alignment
    print(' ' * (n - i - 1), end='')
    # Calculate and print Pascal's triangle numbers
    val = 1
```

```
    for j in range(i + 1):
        print(val, end=' ')
        val = val * (i - j) // (j + 1)
    print()
```

---

15. Zigzag Pattern

```
n = 3
length = 9
for i in range(1, n + 1):
    for j in range(1, length + 1):
        if (i + j) % n == 0:
            print('*', end='')
        else:
            print(' ', end='')
    print()
```
