Questions

1. What will be the output of the following code?

```
i = 1
while i <= 3:
    print(i)
    i += 1
```

2. Write a `for` loop to print each character of the string `"hello"`. What will be the output?

```
for char in "hello":
    print(char)
```

3. How can you use the `range()` function to print numbers from 5 to 9?

4. What will the following code do?

```
for i in range(4):
    if i == 2:
        continue
    print(i)
```

5. Write a `while` loop that prints numbers from 3 to 7. What will be the output?

```
i = 3
while i <= 7:
    print(i)
    i += 1
```

Different Types of Triangle Patterns in Python

1. Right-Angled Triangle

Explanation:

- Use a loop to iterate from 1 to `n` (number of lines).
- Print `i` number of `*` on each line.

2. Inverted Right-Angled Triangle

Explanation:

- Use a loop to iterate from `n` down to 1.
- Print `i` number of `*` on each line.

3. Equilateral Triangle

Explanation:

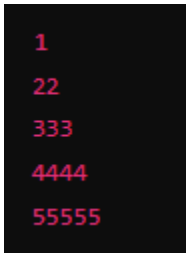
- Use a loop to iterate from 0 to `n-1`.
- Print spaces for padding on the left, then print `*` to form the triangle.
- The number of `*` increases by 2 for each line (1, 3, 5, etc.).

4. Inverted Equilateral Triangle

Explanation:

- Use a loop to iterate from `n` down to 1.
- Print spaces for padding on the left, then print `*` to form the inverted triangle.
- The number of `*` decreases by 2 for each line (9, 7, 5, etc.).

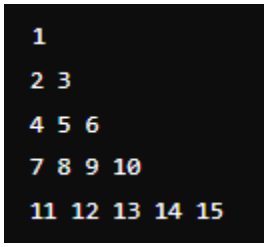
5. Right-Angled Triangle with Numbers



Explanation:

- Use a loop to iterate from 1 to `n`.
- Print `i` number of `i` on each line.

6. Floyd's Triangle

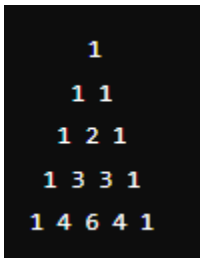


Explanation:

- Use a nested loop.
- Outer loop controls the number of lines.

- Inner loop prints numbers starting from 1 and increasing sequentially.

7. Pascal's Triangle



Explanation:

- Use a nested loop.
- Outer loop for each line.
- Inner loop to calculate and print coefficients using binomial theorem.