**Technical Report: PATTERN PROGRAM**

**Overview:**

Printing a simple pattern like a triangle involves using a single loop to control the number of stars on each line. This print triangle() function prints a triangle pattern where each subsequent row has one additional star.

**Problem Definition:**

A pattern program generates a structured arrangement of characters or numbers based on a predefined logic. The structure can be simple or complex and may involve triangles, rectangles, pyramids, or other geometric shapes.

**Input:** Number of rows or levels (n).

Optional: The character to print (default is usually \*).

**Output:** A visual pattern displayed on the console.

**Algorithm outline:**

The general algorithm to solve a pattern program follows these steps:

1.Take user input (n), which defines the size or number of rows.

2.Use nested loops:

* The outer loop controls the rows.
* The inner loop controls the columns or characters to print in each row.

3.Depending on the pattern type, control the logic inside the inner loop.

4.Print the result.

**Pseudocode**

def print\_triangle(height):

"""

Prints a triangle pattern of asterisks (\*).

Args:

height (int): The height of the triangle.

"""

for i in range(height):

print(' ' \* (height - i - 1) + '\*' \* (2 \* i + 1))

def main():

height = int(input("Enter the height of the triangle: "))

print\_triangle(height)

if \_name\_ == "\_main\_":

main()

def triangle\_pattern(n):

for i in range(1, n+1):

# Print leading spaces for alignment

print(' ' \* (n - i), end='')

# Print the stars

print('\*' \* (2\*i - 1))

# Test the function with 5 rows

n = 5

triangle\_pattern(n)

**Time and Space Complexity:**

Time Complexity: O(n^2) for most patterns, where n is the number of rows, due to the nested loops.

Space Complexity: O(1) if we are printing directly to the console

**Conclusion:**

This documentation provides a general overview for pattern program development. The specific logic can be adapted based on the desired pattern.