

Problem – 2 : Pascal's Triangle

```
def pascals_triangle_element(r, c):
```

```
    if c == 1 or c == r:
```

```
        return 1
```

```
    else:
```

```
        return pascals_triangle_element(r - 1, c - 1) + pascals_triangle_element(r - 1, c)
```

```
def pascals_triangle_row(n):
```

```
    row = []
```

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

```
        row.append(pascals_triangle_element(n, i))
```

```
    return row
```

```
def pascals_triangle(n):
```

```
    triangle = []
```

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

```
        triangle.append(pascals_triangle_row(i))
```

```
    return triangle
```

```
# Variation 1: Print the element at position (r, c) in Pascal's triangle
```

```
r = 5
```

```
c = 3
```

```
element = pascals_triangle_element(r, c)
```

```
print("Result (Variation 1):", element)
```

```
# Variation 2: Print the n-th row of Pascal's triangle
```

```
n = 5
```

```
row = pascals_triangle_row(n)
```

```
print("Result (Variation 2):", ' '.join(str(x) for x in row))
```

```
# Variation 3: Print the first n rows of Pascal's triangle
```

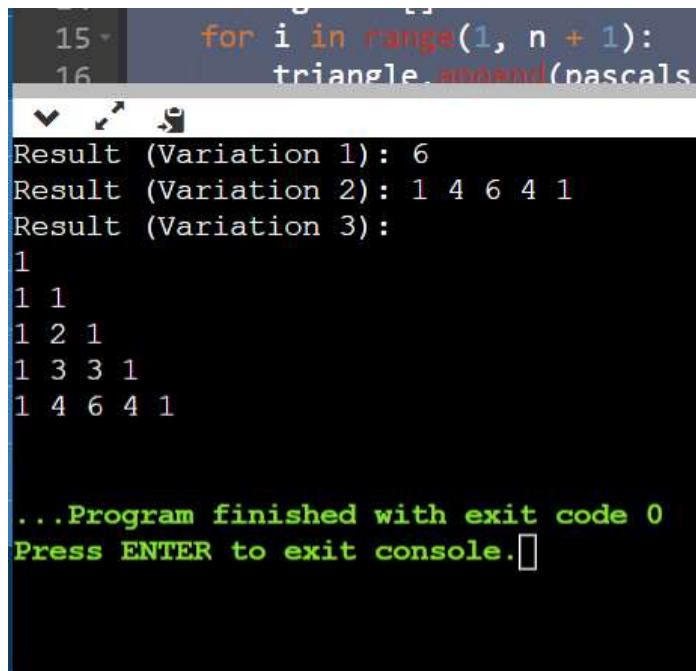
```
n = 5
```

```
triangle = pascals_triangle(n)
```

```
print("Result (Variation 3):")
```

```
for row in triangle:
```

```
    print(' '.join(str(x) for x in row))
```

A screenshot of a Python IDE. The top part shows a code editor with the following code:

```
15 for i in range(1, n + 1):  
16     triangle.append(pascals
```

The bottom part shows a console window with the following output:

```
Result (Variation 1): 6  
Result (Variation 2): 1 4 6 4 1  
Result (Variation 3):  
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```