NAME: OMAR MUDASAR

CMS: 465779

LAB MANUAL: 6

LAB TASK:

1:

```
# Function to generate Fibonacci sequence up to n terms
def generate_fibonacci(n):
    fib_sequence = []

    # Initialize the first two terms
    a, b = 0, 1

    # Generate Fibonacci sequence using a loop
    for _ in range(n):
        fib_sequence.append(a)
        # Update a and b to the next terms in the sequence
        a, b = b, a + b

    return fib_sequence

# Specify the number of terms you want in the Fibonacci sequence
num_terms = 10

# Generate and print the Fibonacci sequence
fibonacci_sequence = generate_fibonacci(num_terms)
print(f"Fibonacci sequence with {num_terms} terms: {fibonacci_sequence}")
```

2:

```
# Function to generate Pascal's triangle up to n rows
def generate_pascals_triangle(n):
    triangle = []

    # Generate Pascal's triangle using nested loops
    for i in range(n):
        row = []
        for j in range(i + 1):
            if j == 0 or j == i:
                # The first and last element in each row is always 1
                row.append(1)
            else:
                # Calculate the middle elements as the sum of the two above
                row.append(triangle[i - 1][j - 1] + triangle[i - 1][j])
        triangle.append(row)

    return triangle

# Specify the number of rows you want in Pascal's triangle
num_rows = 5

# Generate and print Pascal's triangle
pascals_triangle = generate_pascals_triangle(num_rows)

# Print the triangle in a readable format
for row in pascals_triangle:
    print(row)
```

HOME TASKS:

1:

```
def is_prime(num):
    # Function to check if a number is prime
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

# Initialize sum to zero
prime_sum = 0

# Loop through numbers from 1 to 50
for number in range(1, 51):
    # Check if the number is prime
    if is_prime(number):
        # Add the prime number to the sum
        prime_sum += number

# Display the sum of prime numbers
print(f"Sum of prime numbers from 1 to 50: {prime_sum}")
```

2:

```
#include <iostream>

int main() {
   int rows;

   // Input the number of rows for the pattern
   std::cout << "Enter the number of rows: ";
   std::cin >> rows;

   // Loop to create the pattern
   for (int i = 1; i <= rows; ++i) {
      // Loop to print numbers from 1 to i in each row
      for (int j = 1; j <= i; ++j) {
         std::cout << j << " ";
      }

      // Move to the next line after each row
      std::cout << "\n";
   }

   return 0;
}
```

3:

```
#include <iostream>

int main() {
   int rows;

   // Input the number of rows for the pattern
   std::cout << "Enter the number of rows: ";
   std::cin >> rows;

   int num = 1;

   // Loop to create the pattern
   for (int i = 1; i <= rows; ++i) {
      // Loop to print the number 'num' i times in each row
      for (int j = 1; j <= i * 2; ++j) {
         std::cout << num << " ";
      }

      // Double the value of 'num' for the next row
      num *= 2;

      // Move to the next line after each row
      std::cout << "\n";
   }

   return 0;
}
```