AMS 595 / DCS 525: C/C++ Project

Jiacheng Qiu
SBU ID: 117348747

December 10, 2025

1 Work Done

In this project, I implemented several C++ programs to demonstrate proficiency with basic syntax, control structures, and standard library features. The work is organized into five main sections:

1. **Conditional Statements:** I translated a MATLAB `switch` statement into C++ to handle integer cases for -1, 0, and 1.
2. **Vector Printing:** I implemented a helper function `print_vector` using `std::vector` and iterators to display integer arrays in the console.
3. **While Loops:** I wrote a script to generate the Fibonacci sequence. A `while` loop was used to continue generating terms until the value exceeded 4,000,000.
4. **Functions & Prime Analysis:** I developed three functions:
   - `isprime(n)`: Determines if a number is prime by checking divisors up to $\sqrt{n}$.
   - `factorize(n)`: Finds all positive divisors of a number.
   - `prime_factorize(n)`: Computes the prime factorization by iteratively dividing by 2 and subsequent odd numbers.
5. **Pascal's Triangle:** I implemented an iterative approach to generating Pascal's Triangle. Each row was computed based on the sum of adjacent elements from the previous row, stored in dynamic vectors.

2 Results

The following outputs were generated by running the compiled `main.cpp` executable.

Q3: Fibonacci Sequence

The program successfully generated all terms in the Fibonacci sequence starting with 1, 2, and ensuring no term exceeded 4,000,000.

```
--- Q3: Fibonacci Sequence (< 4,000,000) ---
1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765,
    10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040,
    1346269, 2178309, 3524578
```

Q4: Functions and Factorization

The functions correctly identified prime numbers and computed factors for the test cases 2, 10, 17, 72, and 196.

4.1 Prime Check Tests

```
--- Q4.1: isprime Tests ---
isprime(2) = true
isprime(10) = false
isprime(17) = true
```

4.2 Factorization Tests

```
--- Q4.2: factorize Tests ---
Factors of 2: [ 1, 2 ]
Factors of 72: [ 1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72 ]
Factors of 196: [ 1, 2, 4, 7, 14, 28, 49, 98, 196 ]
```

4.3 Prime Factorization Tests

```
--- Q4.3: prime_factorize Tests ---
Prime Factors of 2: [ 2 ]
Prime Factors of 72: [ 2, 2, 2, 3, 3 ]
Prime Factors of 196: [ 2, 2, 7, 7 ]
```

Q5: Pascal's Triangle

The program generated the first 8 rows of Pascal's Triangle using iterative vector summation.

```
--- Q5: Pascal's Triangle (8 rows) ---
[ 1 ]
[ 1, 1 ]
[ 1, 2, 1 ]
[ 1, 3, 3, 1 ]
[ 1, 4, 6, 4, 1 ]
[ 1, 5, 10, 10, 5, 1 ]
[ 1, 6, 15, 20, 15, 6, 1 ]
[ 1, 7, 21, 35, 35, 21, 7, 1 ]
```