AMS 595/DCS 525 – C++ Project 1

Due: November 21th, 2022 - 11:59 PM

NOTE:. *You must write your code on your own. Do not copy code from any source. To get full credit your codes must be well documented with comments. Submit your codes to GitHub* ***and*** *Brightspace along with brief documentation and any relevant plots. Your codes should have the correct file extensions and should run* ***without*** *any needed modifications.*

Q1. Conditional Statements (10 Points)

Translate the following MATLAB code into C++:

```
n = input('Enter a number: ');

switch n
    case -1
        disp('negative one')
    case 0
        disp('zero')
    case 1
        disp('positive one')
    otherwise
        disp('other value')
end
```

Q2. Printing a Vector (10 Points)

There is no built-in function to print a vector in C++, as a result you will have to write your own. Complete the following function that prints a C++ vector of integers. You may call this function later in this assignment.

```
void print_vector(std::vector<int>& v) {
    // Your implementation
}
```

Q3. While Loops (15 Points)

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

$$1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

Using a while loop, return all elements of the Fibonacci sequence whose values do not exceed 4,000,000. Your output does not need to be in the form of an array/vector.

Q4. Functions

Q4a. If Prime (10 Points)

Complete the following function that determines if a number is prime:

```
bool isprime(int n) {
    bool result;
    // Your implementation
    return result;
}
```

In particular, your code should pass the following test cases:

```
void test_isprime() {
    std::cout << "isprime(2) = " << isprime(2) << '\n';
    std::cout << "isprime(10) = " << isprime(10) << '\n';
    std::cout << "isprime(17) = " << isprime(17) << '\n';
}
```

Q4b. Factorize (10 Points)

Complete the following function that finds all the factors of a number:

```
std::vector<int> factorize(int n) {
    std::vector<int> answer;
    // Your implementation
    return answer;
}
```

In particular, your code should pass the following test cases:

```
void test_factorize() {
    print_vector(factorize(2));
    print_vector(factorize(72));
    print_vector(factorize(196));
}
```

Q4c. Prime Factorization (10 Points)

Complete the following function that finds the prime factorization of a number:

```
std::vector<int> prime_factorize(int n) {
    std::vector<int> answer;
    // Your implementation
    return answer;
}
```

In particular, your code should pass the following test cases:

```
void test_prime_factorize() {
    print_vector(prime_factorize(2));
    print_vector(prime_factorize(72));
    print_vector(prime_factorize(196));
}
```

Q5. Recursive Functions and Loops (25 Points)

Write a function that prints the first n rows of Pascal's Triangle. You may use either recursion or iteration. You do not need to format your final solution. You may **not** use combinatorics as part of your solution.

Grading Scheme

- 90 points for correctness of code.
- 10 points for documentation, proper coding practices (commenting, ease of reading, etc.), and correct use of Git.