

Important Note

You must write your code independently. Do not copy the code from any source. To receive full credit, your code must be well-documented with comments. Your script **must** be uploaded to Brightspace. The evaluation will be 10-point scale.

Project Requirements

Due Date

Sunday, 11:59 PM EST, December 14th, 2025

- 1-2 page report, includes:
 - work done
 - results
 - README file contains information about how to run codes
 - Comment or Code Description
- One ZIP file containing commented, executable code

1 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
```

2 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
}
```

3 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, ...

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.

4 Functions

4.1 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';
}
```

4.2 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));  
}
```

4.3 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));  
}
```

5 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.