AMS 595
Xiaoyong Sun
112496580

Q1. Conditional Statements

In this question, translate the "switch" function from MATLAB code into C++ code: in common, the structures of both functions are very similar. Since I might use a few functions multiple times, such as "print_vector()", I also set up "conditional_statements()" as a function. It receives an integer n as input value, split to four different cases (case -1, case 0, case 1, and otherwise). I directly rewrote it to match with four cases. The only details I changed are some notations such as "disp" in MATLAB should represent as "cout<<"statement "<<endl".

Q2. Printing a Vector

Set up a function to display an input vector, it called "print_vector()", inside of this function, I use a for loop with the size of vector to print through each element from the vector. The function is frequently useful and I 'll recall it many times in the following questions.

Q3. While Loops

The fuction "getFibonacci() " will receive a integer called "upper" as a upper bound of a Fibonacci sequence. There are two initial values which are 1 and 2. I use a vector to insert the first values, then go thought a while loop to insert the other values. Each new value is the sum of two previous numbers, collect them until the new element is greater than the upper bound which is 4000,000. Display all elements in the vector.

Q4. Function

a. If prime

In this question, there are two functions, the first function called "isprime()" that will receive an integer as input value, use if-statement to split different cases, if input value n is less than 2, it cannot be prime by the definition, also when n is 2 or 3, it must be prime, the last case is when n is greater than 3, use a for loop to check each element from 2 to $\sqrt{n}+1$, if n is divisible by the element, then n cannot be prime, otherwise n must be a prime.

The second function "test_isprime" is used to recall "isprime" function with certain input n, check if the n is prime or not and display the results.

b. Factorize

In part b, I need to find all the factors for the input value n. Use a for loop to check through all the elements from 1 to n. If n is divisible by the element, then it is one of the factors of n. Collect all the elements into a vector and display. Display the factors for 2, 72 and 196.

c. Prime Factorization

In this part, Prime Factorization also consists of two functions, this first function is "prime_factorize()". It takes an integer n, checks whether it is prime or not first. If n is prime then its prime factorization is itself, such that insert n into vector. Otherwise, n is not prime, then loop through from 2 to n, if the element i is prime and n is divisible by I, then collect the number and keep dividing this number until no more the same factor in n, repeat the process from 2 to n to get all the prime factorization.

The second function "test_prime_factorization()" is call the first function to check all the prime factorization of 2, 72,196, and Display all the factors

Q5. Recursive Functions and Loops

This question needs to be printed out the first n row of Pascal's Triangle. I choose to use recursion to get final solution. I wrote a function, "pascal(m)", to find the mth row coefficient of Pascal's Triangle. It receives the integer m as an input value. If m is 0, then the first row is {1}. Also, if m is not 0, then call the function "pascal(m-1)". Then loop through the first element to the m-1 th element, since each element in the ith row is the sum of the (i-1) th element and ith element from last row, use vector to insert elements in mth row. and inert 1 at end of the row. Return the vector in mth row.

Use the function "print_pascal()" to display the result, loop through from 0 to m and call the function "pascal(m)". print each element in the row, from the first row to mth row.