**Decimal to Roman Numeral Converter**

For this project, I coded using C++ as my language and CLion as my platform.

1. After about an hour, I was able to establish a basic foundation to my code.
   1. To convert from roman to decimal, I accessed each character in the string and then assigned the corresponding decimal value as a component in a vector. I then added all of the components together within a vector. In order for this to correlate with true roman numbers, I had to add the dependence statement that if a number is smaller than a proceeding value, take the difference of those numbers, and then add it to the sum.
   2. To convert from decimal to roman, I used a sequence of while loops which are now commented out under main because I realized that they would only work for decimal numbers which didn’t contain 4 or 9.
2. The second hour of work consisted of trying to solve the problem of how to convert from decimal to a correct format of roman numerals
   1. I started by coding my “createIntVec” which takes the decimal number and puts the value for the thousandth place as the first component, then the hundredth place value as the second component and so on.
   2. I then built the “intToRoman” function which looks at each component of the vector and adds the correct corresponding roman value to a new vector. At the end of this process, the resulting roman number is printed.
3. After getting my code to work as planned, I turned to making sure that it would only accept valid input. This took me almost another full hour.
   1. For the decimal values, I made sure that all inputs were greater than 0 and less than 4,000.
   2. It was a lot harder for the roman numerals, here are the stipulations which I set for a valid roman numeral.
      1. A symbol representing 10x may not precede any symbol larger than 10x+1.
      2. There cannot ever be four of the same roman numerals in a row. For example, it will not accept “CCCC” because this should actually be written as “CD”
      3. There should never be a number which is smaller than one two spots after it. For example, “IIIV” is not valid, it should instead be “VI”
4. After following this whole process, I paused my stopwatch at 2:52:41