```cpp
#include <iostream>
#include <iomanip>
#include <string>
#include <cmath>

using namespace std;


// This function operates by finding the length of a string after removing trailing zeroes and decimal places.
// I wanted to make sure that trailing zeroes are removed, so I had it delete the decimal if it comes across it before a non-zero number.
int sigFigCalculator(string input) {

    while (input[0] == '0' || input[0] == '.') {

        input = input.erase(0, 1);

    }
    // I then wanted to make sure there was no decimal place affecting sig figs, so I used string::npos to ensure that it still exists.
    if (input.find('.') != string::npos) {
        input.erase(input.find('.'), 1);
    }
    // Records and returns length.
    int length = input.length();

    return(length);

}

int main() {

    string loopInput; // Used at the end for our exit loop to read our input
    bool exitLoop; // Used to see if the end loop needs to continue or not
    int loopCheck = 1; // Used to loop the main function until broken
    cout << "Welcome to the Ohm's Law Calculator!";

    while (loopCheck == 1) {

        bool loop; // Used to loop through the calculation process until two proper inputs are given.
        double formulaTrack = 0; // Used to determine which version of V = IR to use depending on variable being requested.
        string inputOne; // First input, stored as a string to determine string length accurately without additional zeroes being added from string to double.
        string inputTwo; // Second input, stored as a string to determine
```

```
            string length accurately without additional zeroes being added
            from string to double.
42          double inputOneCalculation = 0.0; // stod(inputOne);
43          double inputTwoCalculation = 0.0; // stod(inputTwo);
44          double outputCalculation = 0.0; // Calculates the value of the
              given variable based on inputs.
45          int outputOne = 0; // Stores sig figs based on input one.
46          int outputTwo = 0; // Stores sig figs based on input two.
47          int sigFigTrack = 0; // Uses smaller of the two above outputs to
              determine final sig figs.
48
49          cout << endl << "What would you like to calculate?" << endl;
50          cout << "1. Voltage (V)" << endl;
51          cout << "2. Current (I)" << endl;
52          cout << "3. Resistance (R)" << endl;
53          cout << "4. Exit" << endl;
54          cout << "Enter your choice: ";
55          cin >> formulaTrack;
56
57          // This if statement makes sure our choice reflects our options,
              and if a letter is given it checks for a input failure,
              clearing the error and repeats the loop using continue.
58          if (formulaTrack != 1 && formulaTrack != 2 && formulaTrack !=
              3 && formulaTrack != 4 || cin.fail()) {
59              cin.clear();
60              cin.ignore(numeric_limits<streamsize>::max(), '\n');
61              cout << endl << "Error! Please choose a number between 1 and
                  4: \n";
62              continue;
63          }
64
65          if (formulaTrack == 1) {
66              loop = true;
67              while (loop == true) {
68                  // This is effectively our voltage function
69                  cout << endl << "Enter current (I) in Amps : ";
70                  cin >> inputOne;
71                  cout << endl << "Enter resistance (R) in Ohms: ";
72                  cin >> inputTwo;
73
74                  // I found that my inputs would error if they were
                      strings that were non-numbers.
75                  // To fix this I found the try function, which ensures
                      they are numbers, or returns an error.
76                  // In the case there is an error, it outputs it to the
                      terminal and resets the loop using continue.
77                  try {
78                      inputOneCalculation = stod(inputOne);
79                      inputTwoCalculation = stod(inputTwo);
```

```
80                  }
81                  catch (...) {
82                      cout << "\nError! Please reinput your two numbers:
                          \n";
83                      continue;
84                  }
85
86                  // After converting our string to a double, if the double
                      is negative we give an error.
87                  if (inputOneCalculation <= 0 || inputTwoCalculation <= 0)
                      {
88                      cout << endl << endl << "Error! Please reinput your
                          two numbers: \n";
89                  }
90
91                  // If there are no errors, and our values are valid, we
                      can move outside of our variable collection and move on
                      to outputs.
92                  else {
93                      loop = false;
94
95                  }
96              }
97
98              // This calculates the sig figs using the string inputs in
                  our function above
99              // it is done with strings and not doubles as stod would add
                  floating zeroes.
100             outputOne = sigFigCalculator(inputOne);
101             outputTwo = sigFigCalculator(inputTwo);
102             sigFigTrack = min(outputOne, outputTwo);
103
104             // Calculates requested variable based on what we are solving
                  for, in this case, voltage.
105             outputCalculation = (inputOneCalculation *
                  inputTwoCalculation);
106             // Sets our precision to the amount of sig figs gathered
                  above, as in total digits shown.
107             cout << setprecision(sigFigTrack) << "Voltage(V) = " <<
                  outputCalculation << "V" << endl << endl;
108         }
109
110         if (formulaTrack == 2) {
111             loop = true;
112             while (loop == true) {
113                 // This is effectively our current function
114                 cout << endl << "Enter voltage (V) in Volts : ";
115                 cin >> inputOne;
116                 cout << endl << "Enter resistance (R) in Ohms: ";
```

```
117                 cin >> inputTwo;
118
119                 // I found that my inputs would error if they were strings that were non-numbers.
120                 // To fix this I found the try function, which ensures they are numbers, or returns an error.
121                 // In the case there is an error, it outputs it to the terminal and resets the loop using continue.
122                 try {
123                     inputOneCalculation = stod(inputOne);
124                     inputTwoCalculation = stod(inputTwo);
125                 }
126                 catch (...) {
127                     cout << "\nError! Please reinput your two numbers: \n";
128                     continue;
129                 }
130
131                 // After converting our string to a double, if the double is negative we give an error.
132                 if (inputOneCalculation <= 0 || inputTwoCalculation <= 0) {
133                     cout << endl << endl << "Error! Please reinput your two numbers: \n";
134                 }
135
136                 // If there are no errors, and our values are valid, we can move outside of our variable collection and move on to outputs.
137                 else {
138                     loop = false;
139
140                 }
141             }
142
143             // This calculates the sig figs using the string inputs in our function above
144             // it is done with strings and not doubles as stod would add floating zeroes.
145             outputOne = sigFigCalculator(inputOne);
146             outputTwo = sigFigCalculator(inputTwo);
147             sigFigTrack = min(outputOne, outputTwo);
148
149             // Calculates requested variable based on what we are solving for, in this case, voltage.
150             outputCalculation = (inputOneCalculation / inputTwoCalculation);
151             // Sets our precision to the amount of sig figs gathered above, as in total digits shown.
```

```
152             cout << setprecision(sigFigTrack) << "Current(I) = " <<
                  outputCalculation << "A" << endl << endl;
153         }
154
155         if (formulaTrack == 3) {
156             loop = true;
157             while (loop == true) {
158                 // This is effectively our resistance function, notes
                      above apply
159                 cout << endl << "Enter voltage (V) in Volts : ";
160                 cin >> inputOne;
161                 cout << endl << "Enter current (I) in Amps: ";
162                 cin >> inputTwo;
163
164                 // I found that my inputs would error if they were
                      strings that were non-numbers.
165                 // To fix this I found the try function, which ensures
                      they are numbers, or returns an error.
166                 // In the case there is an error, it outputs it to the
                      terminal and resets the loop using continue.
167                 try {
168                     inputOneCalculation = stod(inputOne);
169                     inputTwoCalculation = stod(inputTwo);
170                 }
171                 catch (...) {
172                     cout << "\nError! Please reinput your two numbers:
                          \n";
173                     continue;
174                 }
175
176                 // After converting our string to a double, if the double
                      is negative we give an error.
177                 if (inputOneCalculation <= 0 || inputTwoCalculation <= 0)
                      {
178                     cout << endl << endl << "Error! Please reinput your
                          two numbers: \n";
179                 }
180
181                 // If there are no errors, and our values are valid, we
                      can move outside of our variable collection and move on
                      to outputs.
182                 else {
183                     loop = false;
184
185                 }
186             }
187
188             // This calculates the sig figs using the string inputs in
                  our function above
```

```
189             // it is done with strings and not doubles as stod would add
                  floating zeroes.
190             outputOne = sigFigCalculator(inputOne);
191             outputTwo = sigFigCalculator(inputTwo);
192             sigFigTrack = min(outputOne, outputTwo);
193
194             // Calculates requested variable based on what we are solving
                  for, in this case, voltage.
195             outputCalculation = (inputOneCalculation /
                  inputTwoCalculation);
196             // Sets our precision to the amount of sig figs gathered
                  above, as in total digits shown.
197             cout << setprecision(sigFigTrack) << "Resistance(R) = " <<
                  outputCalculation << endl << endl;
198         }
199         // Removes user from program
200         if (formulaTrack == 4) {
201
202             cout << endl << "Goodbye!" << endl;
203             return(0);
204
205         }
206
207         exitLoop = true; // Used to determine if this final exit program
              loop is done (for when the process above is to be repeated)
208         while (exitLoop == true) {
209
210             cout << "Would you like to perform another calculation? (y/
                  n): ";
211             cin >> loopInput;
212
213             if (loopInput == "y") {
214                 exitLoop = false;
215             }
216             else if (loopInput == "n") {
217
218                 cout << endl << "Goodbye!" << endl;
219                 return(0);
220
221             }
222             else {
223
224                 cout << endl << "Error: Option not included." << endl <<
                      endl;
225
226             }
227         }
228     }
229 }
```

```python
def sigFigCalculator(input):

    # Checks if the leading character of input is a 0 or . and deletes them
    while (input[0] == '0' or input[0] == '.'):

        input = input[1:]

    # In the case that the loop above finds a number before the decimal, we remove any remaining decimals
    input = input.replace(".","")

    # We then record the length (number of sig figs) and return it
    length = len(input)

    return(length)

def main():

    # Loops through this over and over so long as they do not exit the program
    mainLoop = True

    print("Welcome to Ohm's Law Calculator! \n")

    while mainLoop:

        loop = True; # Used within individual formulaTracks to continuously loop until a proper input is given
        formulaTrack = 0 # Determines formula used, 1 for I * R, 2 for V / R, 3 for V / I
        inputOne = "" # Takes a string input
        inputTwo = "" # Takes a string input
        inputOneCalculation = 0.0 # Turns string into double equivalent
        inputTwoCalculation = 0.0 # Turns string into double equivalent
        outputCalculation = 0.0 # Multiplies the doubles above together to find our value
        outputOne = 0 # Finds the sig figs of value inputTwo by running thru sigFigCalculator
        outputTwo = 0 # Finds the sig figs of value inputOne by running thru sigFigCalculator
        sigFigTrack = 0 # Takes the lesser of the two sig figs above to find the final sig figs

        # Asks user what it is calculating
        print("What would you like to calculate? \n")
        print("1. Voltage (V) \n")
        print("2. Current (I) \n")
        print("3. Resistance (R) \n")
```

```
41          print("4. Exit \n")
42
43          # Allows user to respond
44          formulaTrack = eval(input("Enter your choice: "))
45
46          # Checks if user input is equivalent to 1, 2, 3, or 4
47          if formulaTrack in [1,2,3,4]:
48
49              # Loops through this process until it finds two valid inputs,
                  checking for errors each time (letters, <= 0)
50              if formulaTrack == 1:
51                  while loop == True:
52
53                      inputOne = (input("Enter Current (I) in Amps: "))
54                      inputTwo = (input("Enter Resistance (R) in Ohms: "))
55
56                      try:
57                          inputOneCalculation = float(inputOne)
58                          inputTwoCalculation = float(inputTwo)
59
60                      except ValueError:
61                          print("\nError! Please reinput your two numbers:
                              \n")
62                          continue
63
64                      if (inputOneCalculation <= 0 or
                          inputTwoCalculation <= 0):
65
66                          print("Error! Please reinput your two numbers:
                              \n")
67                          continue
68
69                      else:
70
71                          loop = False
72                  # Takes the inputs, runs them thru sigFigCalculator to
                      determine how many sig figs
73                  # and finds the min value between the two to find correct
                      sig figs
74                  outputOne = sigFigCalculator(inputOne)
75                  outputTwo = sigFigCalculator(inputTwo)
76                  sigFigTrack = min(outputOne, outputTwo)
77
78                  # Calculates and then rounds based on sig figs.
79                  outputCalculation = (inputOneCalculation *
                      inputTwoCalculation)
80                  print(f"Voltage(V) = {outputCalculation:.{sigFigTrack}g}
                      V")
81
```

```
82              # Loops through this process until it finds two valid inputs, checking for errors each time (letters, <= 0)
83              elif formulaTrack == 2:
84                      while loop == True:
85
86                          inputOne = (input("Enter Voltage (V) in Volts: "))
87                          inputTwo = (input("Enter resistance (R) in Ohms: "))
88
89                          try:
90                              inputOneCalculation = float(inputOne)
91                              inputTwoCalculation = float(inputTwo)
92
93                          except ValueError:
94                              print("\nError! Please reinput your two numbers: \n")
95                              continue
96
97                          if (inputOneCalculation <= 0 or inputTwoCalculation <= 0):
98
99                              print("Error! Please reinput your two numbers: \n")
100                             continue
101
102                         else:
103
104                             loop = False
105                     # Takes the inputs, runs them thru sigFigCalculator to determine how many sig figs
106                     # and finds the min value between the two to find correct sig figs
107                     outputOne = sigFigCalculator(inputOne)
108                     outputTwo = sigFigCalculator(inputTwo)
109                     sigFigTrack = min(outputOne, outputTwo)
110
111                     # Calculates and then rounds based on sig figs.
112                     outputCalculation = (inputOneCalculation / inputTwoCalculation)
113                     print(f"Voltage(V) = {outputCalculation:.{sigFigTrack}g} A")
114
115             # Loops through this process until it finds two valid inputs, checking for errors each time (letters, <= 0)
116             elif formulaTrack == 3:
117                     while loop == True:
118
119                         inputOne = (input("Enter Volts (V) in Volts: "))
120                         inputTwo = (input("Enter Current (I) in Amps: "))
```

```
121
122                         try:
123                             inputOneCalculation = float(inputOne)
124                             inputTwoCalculation = float(inputTwo)
125
126                         except ValueError:
127                             print("\nError! Please reinput your two numbers: \n")
128                             continue
129
130                         if (inputOneCalculation <= 0 or inputTwoCalculation <= 0):
131
132                             print("Error! Please reinput your two numbers: \n")
133                             continue
134
135                         else:
136
137                             loop = False
138                     # Takes the inputs, runs them thru sigFigCalculator to determine how many sig figs
139                     # and finds the min value between the two to find correct sig figs
140                     outputOne = sigFigCalculator(inputOne)
141                     outputTwo = sigFigCalculator(inputTwo)
142                     sigFigTrack = min(outputOne, outputTwo)
143
144                     # Calculates and then rounds based on sig figs.
145                     outputCalculation = (inputOneCalculation / inputTwoCalculation)
146                     print(f"Voltage(V) = {outputCalculation:.{sigFigTrack}g}")
147
148             elif formulaTrack == 4:
149
150                 print("Goodbye!")
151                 return(0)
152
153             else:
154                 print("Invalid option. Please enter 1, 2, 3, or 4.\n")
155
156         # Loops the closing statement until a valid input is given (y/n)
157         exitLoop = True
158         while(exitLoop == True):
159
160             loopInput = input("Would you like to perform another calculation? (y/n): ")
161
```

```
162             if (loopInput == "y"):
163                 exitLoop = False
164 
165             elif (loopInput == "n"):
166                 print("Goodbye! \n")
167                 return(0)
168 
169             else:
170                 print("Error: Option not included.")
171 
172 
173 main()
```