1. Write a complete `c++` program that will identify triangles. The `main` function should obtain the three lengths as integers and then call your function `void triangle (int a, int b, int c)` which will perform all of the work. If `main` is called with no command line arguments, it will prompt the user for the three side lengths All values must be positive integers. If this is not the case, produce an appropriate error message on `cerr` and exit with return code 1. Otherwise, proceed to step 3.

2. If it is called with 1, 2, or more than 3 arguments, produce an appropriate error message (sent to `cerr`) and exit with return code 1. Otherwise, convert the three arguments to integers (rearranging as appropriate such that the last value is at least as large as the other two values. and proceed with step 3.

3. The `triangle` function is to produce one line of output, listing the three side lengths and one of the following messages (the last value must be at least as large as the other two values, rearrange them as needed). `cout`:

   "Not a triangle"

   "Scalene triangle"

   "Isosceles triangle"

   "Equilateral triangle"

   Where appropriate, prefix the message with the text "Right ". Be certain to return an return code to indicate success at the end of step 3.

4. The `main` function is only allowed to use `cin` and `cerr`. The `triangle` function is only allowed to use `cout`. It is also responsible for rearranging the values `a, b, c` such that `c` is at least as large as the other two lengths.

5. Supply a complete program that exercises your class. (be sure you also exercise the program completely (both the interactive and command line versions, including any boundary conditions). Please save all of your test cases and the output. You will need to add to these cases and run them again with later versions of your program.

   Sample output:

   1 1 1 Equilateral triangle
   3 4 5 Right Scalene triangle
   1 1 9 Not a triangle

You must supply a listing of your program and sample output.