Assignment 2 -- Iterations

Purpose of this task is to write a C++ program which takes a positive real number (called "square" in the following) and tries to find ("guess") the square root (called "root") of this number by repeating a try-and-test algorithm.

Because the square-root function is continuous, we can use the mean value theorem: if we know a lower bound for "root" (lower < root) and an upper bound (upper > root), so the arithmetic mean of lower and upper bounds will provide a guess for the root. The error tolerance (call this variable "error") for this guess is half the difference of upper−lower.

As long as the error is not sufficiently small (start with choosing 0.001) we simply can reduce our guessing intervall by using the current root value as new lower or upper bound for the next pass, depending on whether root*root is greater than square or smaller ("delta" in the subsequent example is the result of root*root−square).

CAUTION! -- the numbers in the following program output are NOT correct, please think for yourself!

What would be a good first lower bound for the number entered? What about the upper bound? Start building and testing your program with a moderate tolerance, and then, after getting trust, you can improve the tolerance value. Remember: the data type double is capable of holding 15 to 16 significant decimal digits.

```
square-root computation by iteration  (by Hans Huber, V-1.23)

enter a positive number: 2.0

pass 1:  lower=1.0      upper=2.0      root=1.0       delta=−1.0
pass 2:  lower=1.0      upper=1.5      root=1.5       delta=0.25
pass 3:  lower=1.25     upper=1.5      root=1.25      delta=−0.4375
pass 4:  lower=1.375    upper=1.5      root=1.375     delta=−0.109375
pass 5:  lower=1.375    upper=1.4375   root=1.4375    delta=0.06640625
pass 6:  lower=1.40625  upper=1.4375   root=1.40625   delta=−0.02246094
pass 7:  lower=1.40625  upper=1.42187  root=1.421875  delta=0.0217285156
pass 8:  lower=1.41406  upper=1.42187  root=1.4140625 delta=−0.000427246

the square root of 2.0 is 1.4140625, with a margin of error 0.0035
```

To develop and also to present your program, an algorithmic sketch would be helpful -- use a structogram for this.