

SOLUTION of (8-2.a):

The C++ code computes $y = \log(x)$ for a given x . The loop in lines 15–18 realizes the Newton iterations for finding the zero of the function

$$f(z) = e^z - x, \quad z \in \mathbb{R}, \quad (8.2.2)$$

where z is the independent variable and x is given. We compute $f'(z) = e^z$ and obtain the Newton iteration

$$z^{(k+1)} = z^{(k)} - \frac{\exp(z^{(k)}) - x}{\exp(z^{(k)})} = z^{(k)} - s, \quad s = 1 - \frac{x}{\exp(z^{(k)})}. \quad (8.2.3)$$

C++ code 8.2.4: The documented function

```
2 double myfunction(double x) {
3     // We require that x > 0.
4     assert(x > 0);
5     // First we factorize x as x = 2^{n+r} = 2^n * x_{new}
6     // with integer n and r ∈ [0.5, 1.5].
7     // Set y = n log(2) and z = log(x_{new}),
8     // then log(x) = y + z.
9     constexpr double dy = 0.693147180559945; // = std::log(2)
10    double y = 0.;
11    // If x = 2^{n+r} with positive n:
12    while (x > 2. * std::sqrt(2.)) {
13        x /= 2.;
14        y += dy;
15    } //
16    // If x = 2^{n+r} with negative n:
17    while (x < std::sqrt(2.)) {
18        x *= 2.;
19        y -= dy;
20    } //
21    // Now, we have found y, and we use Newton iteration for the
22    // function f(z) = exp(z) - x_{new} to find z = log(x_{new}).
23    // Initial guess:
24    double z = x - 1.; //
25    // The update is dz = -f(z)/f'(z)
26    double dz = x * std::exp(-z) - 1.;
27    while (std::abs(dz / z) > std::numeric_limits<double>::epsilon()) {
28        z += dz;
29        dz = x * std::exp(-z) - 1.;
30    }
31    return y + z + dz; //
32 }
```

Get it on 🐱 GitLab (codequiz.hpp).