C++ code 9.4.5: Steffensen's method with logging of iterates

```
template <class Function>
double steffensen_log(Function &&f, double x0,
                      Logger<double> *logger_p = nullptr) {

  double x = x0;
  // TO DO (9-4.c): Modify the function steffensen: use the class Logger to
  // save all the iterations x of the Steffensen's method.
  // START
  bool log_enabled = false;
  if (logger_p != nullptr) {
    log_enabled = true;
    (*logger_p)(x);
  }
  double upd = 1;
  double eps = std::numeric_limits<double>::epsilon();

  while (std::abs(upd) > eps) {
    double fx = f(x); // Only 2 evaluations of f at each step
    if (fx != 0) {
      upd = fx * fx / (f(x + fx) - fx);
      x -= upd;
      if (log_enabled) { (*logger_p)(x); }
    }
  }
  else {
    upd = 0;
  }
}
// END
return x;
}
```