A New Tool for Evaluating and Improving Skill in Code Reading

Daniel Hoffman

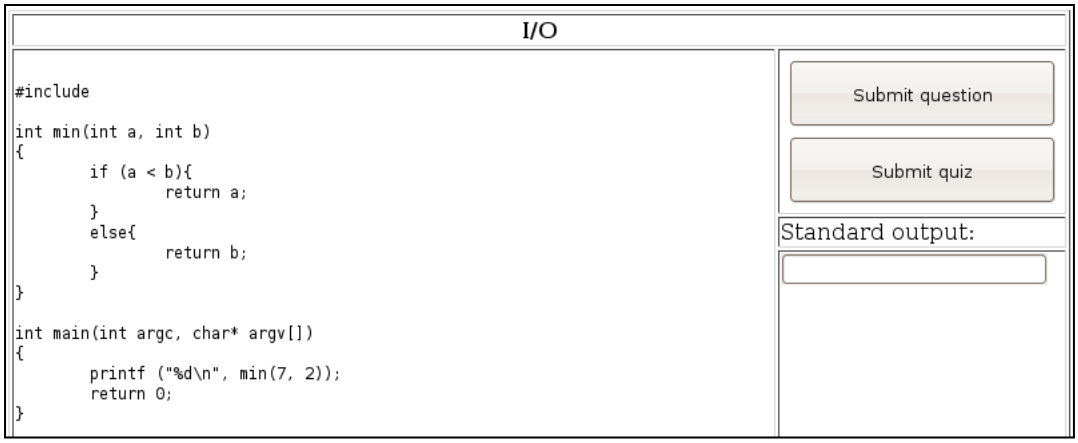
University of Victoria

Learning to write computer software is difficult. You must maintain control over many details and numerous complex relationships. In Computer Science courses, we ask students to write a lot of code. All too often, the resulting code quality is poor: the students have many misunderstandings about their programs and about the details of the programming language they are using.

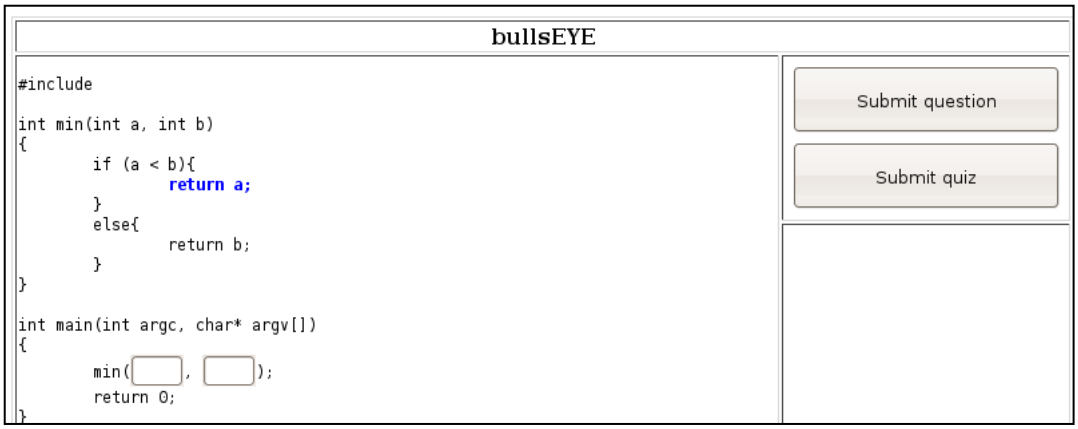
It is widely believed that students would write better code if they spent more time reading code. We already do ask students to read code but the reading tends to be unfocused and little is learned. We lack precise goals and practical evaluation mechanisms for code understanding.

We present a web-based application intended to support evaluation and improvement of code reading skills. The application takes sequence of computer programs, each displayed as a "web form" with a few actual inputs or outputs in the code replaced by input or output fields. The student completes the fields and submits his/her response; the solution is then checked automatically within the application and feedback is provided immediately.

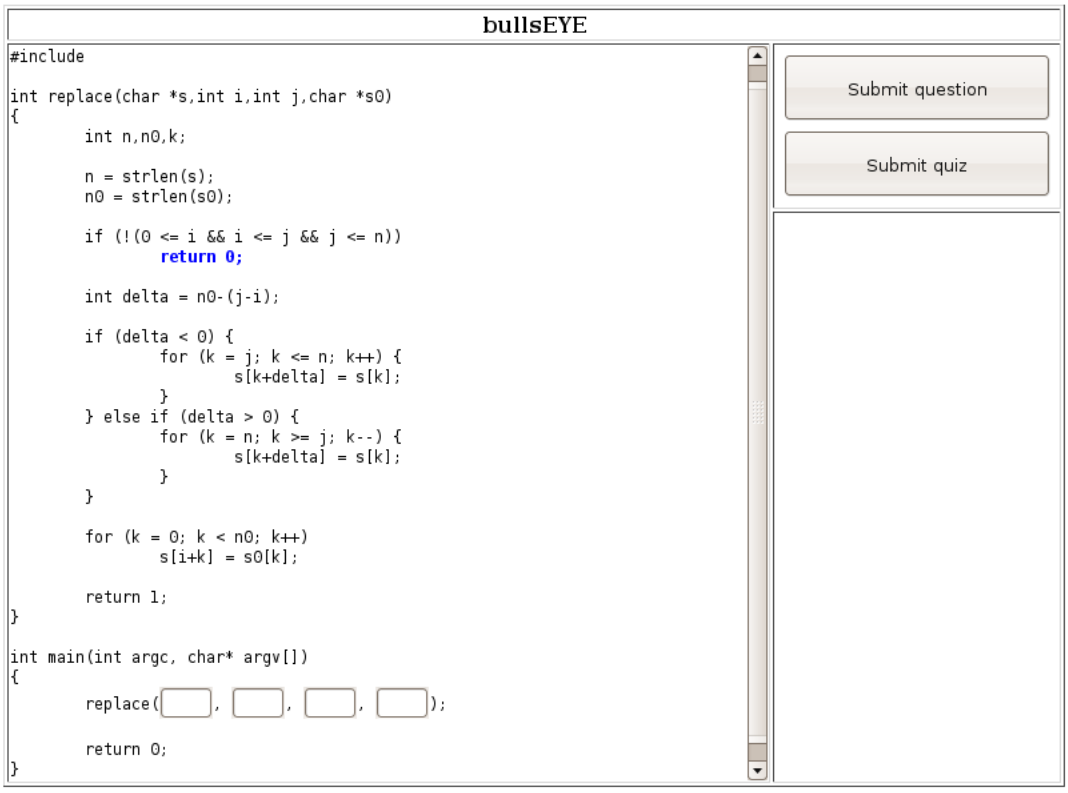
We will consider three examples, all using code in the C programming language. In the first example, the function `min` is defined. The main function issues the call `min(7,2)` and writes the return value to standard output. The student is asked to supply the correct standard output value: 2.



The second example also contains the min function, but the statement `return a;` is highlighted. The student is asked to provide values for the parameters to min. Any values will do as long as, when `min` is invoked, the target statement is executed. Among the correct answers are `min(2,3)`, `min(2,4)`, and `min(-1000,1000)`. The answer `min(2,1)` is incorrect.



While the previous two examples are trivial to solve, there is no limit to the complexity of the code examples. In the third example, the student must provide values for the parameters to `replace` so that the statement `return 0;` is executed.



The full web-based system contains:

- a substantial library of questions,
- support for authoring new questions, and
- logging of quiz results for evaluation and grading purposes.

The system is now in use in a second-year Software Engineering course.

At WTST, I plan a short presentation: five minutes of motivation followed by a 10-minute demonstration.

While the focus of this system is code reading there is a strong testing connection. The bullsEYE questions are exercises in code coverage, focusing on one line at a time. More important, the entire approach is focused on testing: each question is based directly on a test case: concrete inputs, outputs, and execution paths.

I think that the WTST participants would find the tool interesting. I predict that the discussion will be animated. I would very much like to hear that discussion.