```
<html>
    <head>
        <title>Integrader &raquo; Various Practice Problems</title>
        <link rel="stylesheet" href="/css/reset.css" />
        <link rel="stylesheet" href="/css/style.css" />
        <script>var __adobewebfontsappname__ = "code"</script>
        <script>var __adobewebfontsappname__ = "code"</script>
        <script src="http://use.edgefonts.net/droid-sans-
mono:n4:all;raleway:n1,n7,n4,n5,n3,n8,n9,n2,n6:all.js"></script>

        <script src="/js/jquery.js"></script>

        <script src="/js/inputParser.js"></script>

        <script type="text/x-mathjax-config">
        // MathJax Config
MathJax.Hub.Config({
  tex2jax: {inlineMath: [['$$','$$'], ['\\(','\\)']]},
  messageStyle: "none",
  menuSettings: {
    context: "Broswer"
  },
  TeX: {
    extensions: ["color.js"]
  }
});
        </script>
        <script type="text/javascript"
src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML">
</script>
    </head>
    <body>
        <div class="latex-init">
            $$ \usepackage{color} \usepackage[names]{xcolor} $$
        </div>
        <header>
            <div class="logo"><span class="inte">inte</span><span
class="grader">grader</span></div>
            <h1 class="assignment-title"></h1>
        </header>
        <div class="unselectable message">Press enter again to submit.</div>
        <div class="sidebar">
            <ol class="sidebar-items"></ol>
        </div>
        <div class="content">
            <div class="question-area"></div>
            <div class="answer-area"></div>
        </div>

        <script>
            // Submit logic - press enter twice to submit
            var submitTimeout = null;
            $("body").keypress(function(data){
                if(data.keyCode == 13){ // Enter key
                    if(submitTimeout){
                        submit();
                        $(".message").text("Answers submitted.");
                        setTimeout(function(){
                            $(".message").css("opacity", 0);
                            submitTimeout = null;
                        }, 1000);
                        submitTimeout = null;
                    } else {
                        $(".message").text("Press enter again to submit all changed 
answers.");
                        $(".message").css("opacity", 1);
                        submitTimeout = setTimeout(function(){
                            $(".message").css("opacity", 0);
                            submitTimeout = null;
                        }, 5000);
                    }
                }
            });

            // Code to actually run on submit
            var submit = function(){
                console.log("Submitted.");
                $(".changed").each(function(){ // Run on all changed answers
                    this.lastValue = $(this).val();
                    $(this).removeClass("changed");

                    // Run the request
                    req = new XMLHttpRequest();
                    req.open("POST","/answers",false)
                    req.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
                    question = $(this).parent().parent().attr("id");
                    part = $(this).parent().attr("id");
                    answer = $(this).val();
                    req.send(JSON.stringify({question: question, part: part, answer: 
answer}));
                    output = req.response === "true";

                    // Set the new colors
                    if(output){
                        $("#" + question + " #" + part).children("input, textarea, 
select").addClass("correct").attr("disabled", true);
                    } else {
                        $("#" + question + " #" + part).children("input, textarea, 
select").addClass("incorrect");
                    }

                    updateProgress();
                });
            }

            // Loading logic

            var containers = [];
            var loadSet = function(problems){ // Load the entire set
                $("h1.assignment-title").text(problems.title);
                for(p = 0; p < problems.questions.length; p++){
                    item = $('<li><div class="progress-area"><div class="correct">
</div><div class="locked"></div><div class="incorrect"></div></div>' + 
problems.questions[p].title + '</li>');
                    $("ol.sidebar-items").append(item);
                    item[0].number = p;
                    el = $(loadQuestion(problems.questions[p], p));
                    containers.push(el);
                }
            }
            var loadQuestion = function(problem, num){ // Load a single question
                questions = $('<ol class="questions"></ol>');
                answers = $('<ol id="' + num + '" class="answers"></ol">');

                questions.append('<h2 class="question-title">' + problem.title +
'</h2>');

                pcount = 0;

                for(i = 0; i < problem.content.length; i++){
                    if(typeof(problem.content[i]) == "string"){ // Text paragraph
                        questions.append("<p>" + problem.content[i] + "</p>");
                    } else { // Actual question
                        questions.append("<li>" + problem.content[i].text + "</li>");
                        switch(problem.content[i].type){
                            case "input":
                                el = '<input type="text" placeholder="Enter a
number..." />';
                                break;
                            case "select":
                                el = '<select><option value="0" disabled
selected>Choose an answer...</option>';
                                for(j = 0; j < problem.content[i].options.length;
j++){
                                    el += '<option value="' + j + '">' +
problem.content[i].options[j] + "</option>";
                                }
                                el += "</select>";
                                break;
                            case "check":
                                el = '';
                                for(j = 0; j < problem.content[i].options.length;
j++){
                                    el += '<input type="checkbox" id="' + i + "-" + j
+ '" value="' + j + '" /><label for="' + i + "-" + j + '">' +
problem.content[i].options[j] + "</label><br />";
                                }
                                break;
                            case "symbolic":
                                el = '<textarea class="symbolic-input"
placeholder="Enter an expression..."></textarea><div class="symbolic-output"></div>';
                                el = $(el);
                                el[0].vars = problem.content[i].variables;
                                break;
                        }
                        elm = $("<li id='" + pcount + "'></li>").append(el);
                        answers.append(elm);
                        pcount++;
                    }
                }
                return [questions, answers];
            }

            // Question switching logic
            var currentQuestion = -1;
            var switchQuestion = function(number){
                if(number == currentQuestion){
                    return;
                }
```

```
        // Switch sidebar asthetics
        $("ol.sidebar-items li").removeClass("active");
        $($("ol.sidebar-items li")[number]).addClass("active");

        // Swap out question number
        if(currentQuestion != -1){
            containers[currentQuestion][0] = $("ol.questions").detach();
            containers[currentQuestion][1] = $("ol.answers").detach();
        }
        $("div.question-area").append(containers[number][0]);
        $("div.answer-area").append(containers[number][1]);

        currentQuestion = number;

        init(); // Reinit
      }

      // Checks to see if the value of an input has changed
      var checkForChange = function(data){
        console.log($(this).val() + "/ /" + this.lastValue);
        if($(this).val() != this.lastValue){
            $(this).addClass("changed");
        } else {
            $(this).removeClass("changed");
        }
      }

      // Initialize
      var init = function(data){
        $("input, textarea, 
select").keyup(checkForChange).change(checkForChange);
        $("select[type='checkbox']").click(checkForChange);

        $(".symbolic-input").keyup(function(data){
            latexed = inputToLatex($(this).val(), this.vars)
            $(this).parent().children(".symbolic-
output").text(latexed.latexString);
            console.log(latexed.errors);

MathJax.Hub.Queue(["Typeset",MathJax.Hub,$(this).parent().children(".symbolic-
output")[0]]);
        });

        $("textarea").focusin(function(data){
            $(this).parent().children(".symbolic-output").css("visibility", 
"visible").css("opacity", 1);
        });

        $("textarea").focusout(function(data){
            $(this).parent().children(".symbolic-output").css("opacity", 0);
            setTimeout(function(){
                $(this).parent().children(".symbolic-
output").css("visibility", "hidden");
            }, 300);
        });

        $("*").keypress(function(data){
            if(data.keyCode == 13){
                data.preventDefault();
            }
        });
```

```
                $("ol.sidebar-items li").click(function(data){
                    switchQuestion(this.number);
                });

                MathJax.Hub.Queue(["Typeset", MathJax.Hub]);
                MathJax.Hub.Queue(realign);
            }

            // Align question and answer spots
            var realign = function(){
                var min = 0;
                $("ol.questions li").each(function(index){
                    pos = Math.max($(this).position().top + 16, min);
                    if(pos == min){
                        $(this).css("margin-top", (min-$(this).position().top) +
"px");
                    }
                    el = $($("ol.answers li")[index]).css("top", pos);
                    $(this).height(el.height());
                    min = el.position().top + el.outerHeight(true);
                })
                $(".answer-area").height($(".question-area .questions").height());
            }

            // Update progress sliders
            var updateProgress = function(){
                correct = 0;
                incorrect = 0;
                locked = 0;
                $("ol.answers li>*").each(function(num, elem){
                    if($(elem).hasClass("correct")){
                        correct++;
                    }
                    if($(elem).hasClass("incorrect")){
                        correct++;
                    }
                    if($(elem).hasClass("locked")){
                        correct++;
                    }
                });
                $("ol.sidebar-items li.active div.progress-area
div.correct").css("width", correct*100/$("ol.answers li").length + "%");
                $("ol.sidebar-items li.active div.progress-area
div.incorrect").css("width", incorrect*100/$("ol.answers li").length + "%");
                $("ol.sidebar-items li.active div.progress-area
div.locked").css("width", locked*100/$("ol.answers li").length + "%");
            }

            // Gets URL query variable
            getQueryVariable = function(variable){
                var query = window.location.search.substring(1);
                var vars = query.split("&");
                for (var i=0;i<vars.length;i++) {
                    var pair = vars[i].split("=");
                    if(pair[0] == variable){return pair[1];}
                }
                return(false);
            }

            window.onresize = realign;
            // Initialize
            document.ready = function(){
                var req = new XMLHttpRequest();
                req.open("GET","/courses",false);
                req.send();

                courseList = JSON.parse(req.response);

                courseId = getQueryVariable("course");
                assignmentId = getQueryVariable("assignment");

                var problems = {};

                for(i = 0; i < courseList.length; i++){
                    if(courseList[i].UID == courseId){
                        for(j = 0; j < courseList[i].assignments.length; j++){
                            if(courseList[i].assignments[j].UID == assignmentId){
                                problems = courseList[i].assignments[j];
                                break;
                            }
                        }
                        break;
                    }
                }
                loadSet(problems);
                switchQuestion(0);
            }
        </script>
    </body>
</html>
```