JavaScript

```
var x;

// sameDerivative checks to see if the derivative with respect to x and C are equal.
sameDerivative = function(a,b) {
    return (a.derivative('x').equals( b.derivative('x') ) && a.derivative('C').equals( b
};


function factorCheck(f,g) {
    // This validator is designed to check that a student is submitting a factored polyn
    //  Checking that there are the correct number of non-numeric and non-inverse factor
    //  Checking that the submitted answer and the expected answer are the same via real
    //  Checking that the outer most (last to be computed when following order of operat

    var operCheck = f.tree[0];// Check to see if the root operation is multiplication at
    var studentFactors = f.tree.length;// Temporary number of student-provided factors (

    // Now we adjust the length to remove any numeric factors, or division factors, etc
    for (var i = 0; i < f.tree.length; i++) {
        if ((typeof f.tree[i] === 'number')||(f.tree[i][0] == '-')||(f.tree[i][0] == '/
            studentFactors = studentFactors - 1;
        }
    }

    // Now we do the same with the provided answer, in case sage or something provides a
    var answerFactors = g.tree.length;

    // Adjust length in the same way, so that it will match the students if it should.
    for (var i = 0; i < g.tree.length; i++) {
        if (typeof g.tree[i] === 'number') {
            answerFactors = answerFactors - 1;
        }
    }

    // Note: An especially dedicated student could pad with weird factors that all happe
    // For example, a student could enter sin^2(x)+cos^2(x) as a multiplicative factor t
    // This would be somewhat difficult to think of, even on purpose.
    // Until I can reliably evaluate the factors themselves as functions though, there i

    return ((f.equals(g))&&(studentFactors==answerFactors)&&(operCheck=='*'))
}

```

Problem 1 *Factor the following radicand to force the type two radical into* *being a type one radical:*

$$\sqrt{25\,x^3+135\,x^2+216\,x+108}=\sqrt{\boxed{(5\,x+6)(5\,x+6)(x+3)}}$$

Feedback(*`attempt`*): *You should have all linear factors above. In particular you should not have any factors higher than degree 1 polynomials. Essentially you can treat the radicand as if it were a polynomial factoring problem and factor the polynomial just like you would have normally.*

Note: *Repeated factors should be entered in distinctly for the validator. For example, instead of writing something like* $x^2(x-1)^2$ *you should enter it in as* $(x-0)(x-0)(x-1)(x-1)$.

Problem 2 *Factor the following radicand to force the type two radical into being a type one radical:*

$$\sqrt{4\,x^2-8\,x-12}=\sqrt{\boxed{(2\,x+2)(2\,x-6)}}$$

Feedback(*`attempt`*): *You should have all linear factors above. In particular you should not have any factors higher than degree 1 polynomials. Essentially you can treat the radicand as if it were a polynomial factoring problem and factor the polynomial just like you would have normally.*

Note: *Repeated factors should be entered in distinctly for the validator. For example, instead of writing something like* $x^2(x-1)^2$ *you should enter it in as* $(x-0)(x-0)(x-1)(x-1)$.

Problem 3 *Factor the following radicand to force the type two radical into being a type one radical:*

$$\sqrt{36\,x^4+108\,x^3-576\,x^2+432\,x}=\sqrt{\boxed{(3\,x)\,(4\,x-4)(3\,x-6)(x+6)}}$$

Feedback(*`attempt`*): *You should have all linear factors above. In particular you should not have any factors higher than degree 1 polynomials. Essentially you can treat the radicand as if it were a polynomial factoring problem and factor the polynomial just like you would have normally.*

Note: *Repeated factors should be entered in distinctly for the validator. For example, instead of writing something like* $x^2(x-1)^2$ *you should enter it in as* $(x-0)(x-0)(x-1)(x-1)$.