```
var mathjs = require("mathjs");
var math = mathjs();
var steps =4;
var variables = ["a_0","b","beta"];
var range = [[2,5],[5,8],[8,11]];
var eq1 = "a_0(b)/beta";
var eq2="a_0*(b)/beta";

function compare(equation1,equation2,variables,values)
{
        var instructions1 =[];
        var instructions2 =[];
        var len=variables.length;
        for(var i = 0; i<len;i++)
        {
                instructions1.push(variables[i]+"="+values[i]);
                instructions2.push(variables[i]+"="+values[i]);
        }
        instructions1.push("round("+implicitMultiplication(equation1)+",3)");
        instructions2.push("round("+implicitMultiplication(equation2)+",3)");
        console.log((math.eval(instructions1)[len]+", "+(math.eval(instructions2)
[len])))
        return ((math.eval(instructions1)[len])==(math.eval(instructions2)[len]))
}
function test(eq1,eq2,variables,range,steps,array,index)
{

        if(array.length==variables.length)
        {
                return compare(eq1,eq2,variables,array);
        }
        for(var i=range[index][0];i<range[index][1];i+=(range[index][1]-range[index]
[0])/(steps-1))
        {

if(!test(eq1,eq2,variables,range,steps,array.slice(0).concat([i]),index+1))
                        return false;
        }
        return true;
}
//console.log(test(eq1,eq2,variables,range,4,[],0)); //Example Check
function singleVal(equation,variables,values,answer,tolerance)
{
        instructions=[];
        var len=variables.length;
        for(var i = 0; i<len;i++)
        {
                instructions.push(variables[i]+"="+values[i]);
        }
        instructions.push(implicitMultiplication(equation));
        var trueValue = math.eval(instructions)[len];
        var testValue = math.eval(implicitMultiplication(answer));
        return(testValue>=trueValue*(1-tolerance)&&testValue<=trueValue*
(1+tolerance))
}
function getAllIndexes(str)
{
        var openParens=[];
        var check = str.indexOf('(')
        while(check!=-1)
```

```
        {
                openParens.push(check);
                check = str.indexOf('(',check+1)
        }
        var closeParens=[];
        check = str.indexOf(')')
        while(check!=-1)
        {
                closeParens.push(check);
                check = str.indexOf(')',check+1)
        }
        return [openParens,closeParens];
}
function implicitMultiplication(str) //appends '*' as necessary to run math.js with
implicit multiplication
{
        var indexes=getAllIndexes(str)
        var offset = 0;
        var i;
        for(var x in indexes[0])
        {
                i=indexes[0][x];
                if((i-1+offset)>-1 &&
"0123456789)abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVQXYZ".indexOf(str.charAt(
i-1+offset))>-1)
                {
                        str=str.substring(0,
(i+offset))+"*"+str.substring((i+offset),str.length)
                        offset+=1;
                }
        }
        for(var x in indexes[1])
        {
                i=indexes[1][x];
                if((i+1+offset)<str.length &&
"0123456789)abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVQXYZ".indexOf(str.charAt(
i+1+offset))>-1)
                {
                        str=str.substring(0,
(i+offset+1))+"*"+str.substring((i+offset+1),str.length)
                        offset+=1;
                }
        }
        return str;
}

module.exports = {
        "test": test,
        "singleVal": singleVal,
}

//console.log(singleVal("a(b)",["a","b"],[100,1],"1*100",.02)) //Example Check
//console.log(implicitMultiplication("a(b)"))
/*function sigfig(equation,variables,values,answer,degree)
{
        instructions=[];
        var len=variables.length;
        for(var i = 0; i<len;i++)
        {
                instructions.push(variables[i]+"="+values[i]);
        }
```

```
        instructions.push(equation);
        var trueValue = new Number(math.eval(instructions)[len]);
        var str = trueValue.toPrecision(degree);
        if(str.indexOf(".")==-1)
        {
                if(str.substr(str.length-1)==0)
                {

                }
        }
}*/
```