```
operators = ["+", "-", "*", "/", " ", "^", "_"];
leftParens = ["(", "[", "{", "|"];
rightParens = [")", "]", "}", "|"];
functions = ["sin", "cos", "tan", "cot", "sec", "csc", "ln", "log"];
firstSeparators = ["hat", "vector"];

replacements = {
    "theta": "\\theta",
    "phi": "\\phi",
    "Theta": "\\Theta",
    "Phi": "\\Phi",
    "mu": "\\mu",
    "Mu": "\\Mu",
}

var seps = [];
var vars = [];

var inputToLatex = function(input, variables){
    var arr = [input];
    var parenStack = [];
    err = [];

    //separate operators and parens

    variables.sort(function(a, b){
        return b.length - a.length;
    });

    seps =
[].concat(firstSeparators).concat(variables).concat(operators).concat(leftParens).con
cat(rightParens).concat(functions);
    vars = variables;

    for(var i = 0; i < seps.length; i++){
        var arr2 = [];
        for(var j = 0; j < arr.length; j++){
            if(arr[j].charAt(0) == "\u0000"){
                arr2.push(arr[j]);
                continue;
            }
            arr3 = arr[j].split(seps[i]);
            for(var k = 0; k < arr3.length; k++){
                arr2.push(arr3[k]);
                if(k < arr3.length-1){
                    arr2.push("\u0000" + seps[i]);
                }
            }
        }
        arr = arr2;
    }

    //separate out numbers

    for(var i = 0; i < arr.length; i++){
        for(var j = arr[i].length; j > 0; j--){
            if(!isNaN(arr[i].substring(0, j))){
                arr.splice(i, 1, +arr[i].substring(0, j), arr[i].substring(j));
                break;
            }
        }
    }

    //cleanup empty spots and null characters

    arr2 = []

    for(var i = 0; i < arr.length; i++){
        if(typeof(arr[i]) != "string"){
            arr2.push(arr[i]);
            continue;
        }
        if(arr[i].charAt(0) == "\u0000"){
            arr[i] = arr[i].substring(1);
        }
        if($.trim(arr[i]).length > 0){
            arr2.push(arr[i]);
        }
    }

    arr = arr2;

    //combine hats and vectors

    for(var i = 0; i < arr.length; i++){
        if(arr[i] == "hat" && i > 0){
            if(arr[i-1] == "i" || arr[i-1] == "j"){
                arr[i-1] = "\\" + arr[i-1] + "math";
            }
            arr[i-1] = "\\widehat{" + arr[i-1] + "}";
            arr.splice(i, 1);
            i--;
        }

        if(arr[i] == "vector" && i > 0){
            if(arr[i-1] == "i" || arr[i-1] == "j"){
                arr[i-1] = "\\" + arr[i-1] + "math";
            }
            arr[i-1] = "\\vec{" + arr[i-1] + "}";
            arr.splice(i, 1);
            i--;
        }
    }

    //markup variables

    for(var i = 0; i < arr.length; i++){
        if(seps.indexOf(arr[i]) == -1 && variables.indexOf(arr[i]) == -1 &&
isNaN(arr[i]) && !(i > 0 && arr[i-1] == "log" && arr[i].charAt(0) == "_")){
            err.push("Unknown variable error: \"" + arr[i] + "\" is not a variable
given in the problem.");
            arr[i] = "\\textcolor{red}{" + arr[i] + "}"
        }
    }

    //group parens

    for(var i = 0; i < arr.length; i++){
        index = leftParens.indexOf(arr[i]);
        if(index >= 0 && !(arr[i] == "|" && parenStack.length > 0 &&
parenStack[parenStack.length-1][0] == index)){
```

```
            parenStack.push([index, i]);
        } else {
            index = rightParens.indexOf(arr[i]);
            if(index >= 0){
                if(parenStack.length > 0 && parenStack[parenStack.length-1][0] ==
index){
                    arr.splice(parenStack[parenStack.length-1][1], 2,
arr.splice(parenStack[parenStack.length-1][1]+1, i-1-parenStack[parenStack.length-1]
[1]));
                    info = parenStack.pop();
                    i = info[1];
                    if(leftParens[index] != "{"){
                        arr[i] = ["\\left" +
leftParens[index]].concat(arr[i]).concat(["\\right" + rightParens[index]]);
                    } else {
                        arr[i] = ["\\left\\{"].concat(arr[i]).concat(["\\right\\}"]);
                    }
                } else {
                    err.push("Mismatched paren error: \"" + arr[i] + "\" has no
matching left paren.");
                    if(rightParens[index] != "}"){
                        arr[i] = "\\textcolor{red}{" + rightParens[index] + "}";
                    } else {
                        arr[i] = "\\textcolor{red}{\\}}";
                    }
                }
            }
        }
    }

    while(parenStack.length > 0){
        info = parenStack.pop();
        err.push("Mismatched paren error: \"" + leftParens[info[0]] + "\" has no
matching right paren.");
        if(leftParens[info[0]] != "{"){
            arr[info[1]] = "\\textcolor{red}{" + leftParens[info[0]] + "}";
        } else {
            arr[info[1]] = "\\textcolor{red}{\\{}";
        }
    }

    console.log(arr);

    //parse

    parse(arr);

    return {
        latexString: "$$" + arr.join(" ") + "$$",
        errors: err
    };
}

var parse = function(arr){
    if(typeof(arr) == "string"){
        arr = [arr];
    }

    for(var i = 0; i < arr.length; i++){
        index = functions.indexOf(arr[i]);
        if(index != -1){
            arr[i] = "\\" + arr[i];
            if(arr[i] == "\\log" && i < arr.length-2 && typeof(arr[i+1]) == "string"
&&  arr[i+1].charAt(0) == "_"){
                arr[i] = [arr[i], "_{", arr[i+2], "}"];
                arr.splice(i+1, 2);
            }
            if(i < arr.length-1 && !(seps.indexOf(arr[i+1]) >= 0 &&
vars.indexOf(arr[i+1]) < 0)){
                arr.splice(i, 2, arr.slice(i, i+2));
            } else {
                err.push("Missing argument error: \"" + arr[i] + "\" requires an
argument but none is given.");
                arr[i] = ["\\textcolor{red}{", arr[i], "}"];
            }
        }
    }

    for(var i = 0; i < arr.length; i++){
        if(typeof(arr[i]) == "object"){
            arr[i] = parse(arr[i]);
        }
        if(arr[i] == "*"){
            arr[i] = "\\times";
        }
        if(replacements.hasOwnProperty(arr[i])){
            arr[i] = replacements[arr[i]];
        }
    }

    for(var i = 0; i < arr.length; i++){
        if(arr[i] == "/"){
            arr2 = arr.splice(i-1, 3);
            arr.splice(i-1, 0, ["\\frac{", arr2[0], "}{", arr2[2], "}"]);
            i--;
        }
        if(arr[i] == "^"){
            arr2 = arr.splice(i-1, 3);
            arr.splice(i-1, 0, ["{", arr2[0], "}^{", arr2[2], "}"]);
            i--;
        }
    }

    for(var i = 0; i < arr.length; i++){
        if(typeof(arr[i]) == "object"){
            arr[i] = arr[i].join(" ");
        }
    }

    return arr.join(" ");
}
```