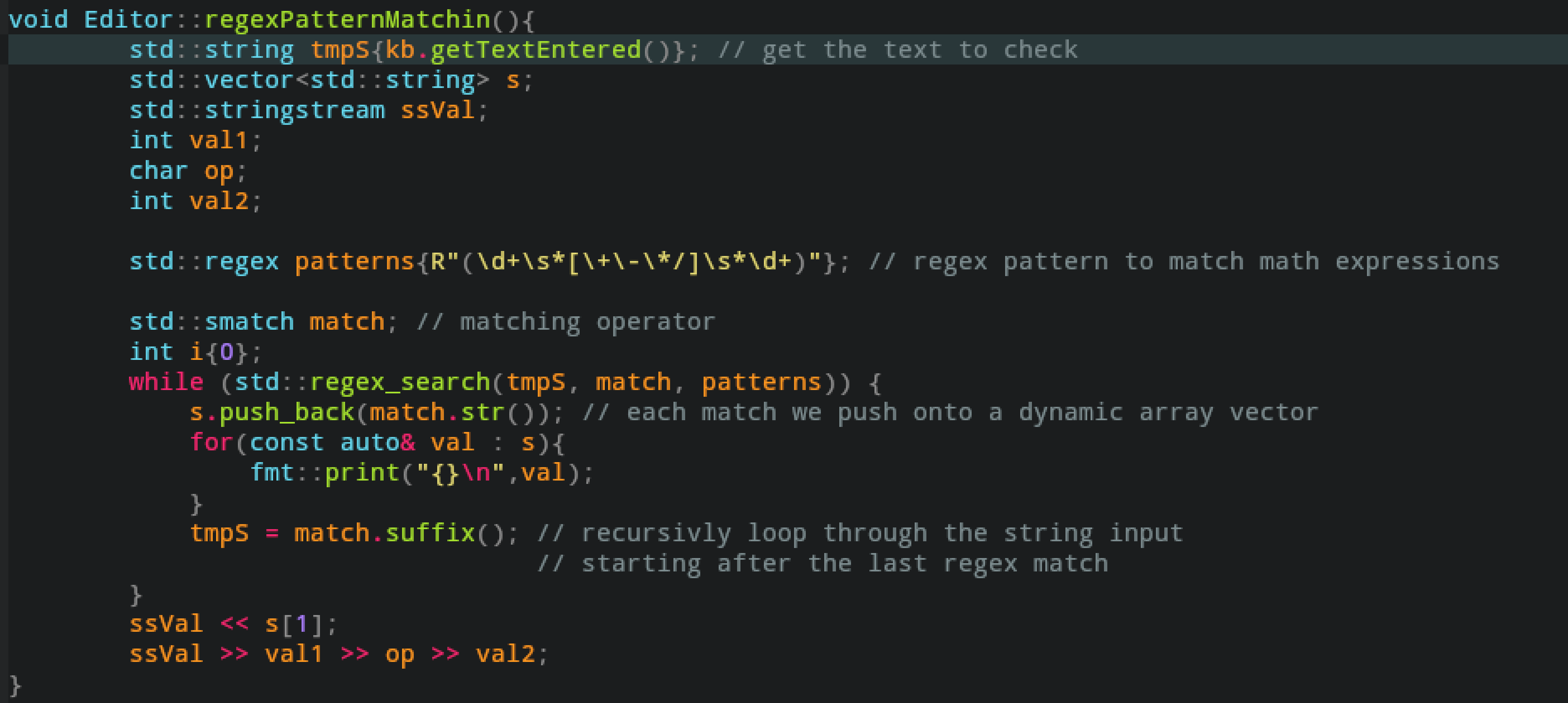

```
void Editor::regexPatternMatchin(){
        std::string tmpS{kb.getTextEntered()}; // get the text to check
        std::vector<std::string> s;
        std::stringstream ssVal;
        int val1;
        char op;
        int val2;

        std::regex patterns{R"(\d+\s*[\+\-\*/]\s*\d+)"}; // regex pattern to match math expressions

        std::smatch match; // matching operator
        int i{0};
        while (std::regex_search(tmpS, match, patterns)) {
            s.push_back(match.str()); // each match we push onto a dynamic array vector
            for(const auto& val : s){
                fmt::print("{}\n",val);
            }
            tmpS = match.suffix(); // recursivly loop through the string input
                                   // starting after the last regex match
        }
        ssVal << s[1];
        ssVal >> val1 >> op >> val2;
}
```