```
class Solution {
    public List<String> removeComments(String[] source) {
        List<String> res = new ArrayList<>();
        StringBuilder sb = new StringBuilder();
        boolean mode = false;
        for (String s : source) {
            for (int i = 0; i < s.length(); i++) {
                if (mode) {
                    if (s.charAt(i) == '*' && i < s.length() - 1 && s.charAt(i + 1) == '/') {
                        mode = false;
                        i++;        //skip '/' on next iteration of i
                    }
                }
                else {
                    if (s.charAt(i) == '/' && i < s.length() - 1 && s.charAt(i + 1) == '/') {
                        break;      //ignore remaining characters on line s
                    }
                    else if (s.charAt(i) == '/' && i < s.length() - 1 && s.charAt(i + 1) == '*') {
                        mode = true;
                        i++;            //skip '*' on next iteration of i
                    }
                    else
                        sb.append(s.charAt(i));     //not a comment
                }
            }
            if (!mode && sb.length() > 0) {
                res.add(sb.toString());
                sb = new StringBuilder();   //reset for next line of source code
            }
        }
        return res;
    }
}
```

The string `//` denotes a line comment, which represents that it and rest of the characters to the right of it in the same line should be ignored.

The string `/*` denotes a block comment, which represents that all characters until the next (non-overlapping) occurrence of `*/` should be ignored. (Here, occurrences happen in reading order: line by line from left to right.) To be clear, the string `/*/` does not yet end the block comment, as the ending would be overlapping the beginning.

The first effective comment takes precedence over others: if the string `//` occurs in a block comment, it is ignored. Similarly, if the string `/*` occurs in a line or block comment, it is also ignored.

If a certain line of code is empty after removing comments, you must not output that line: each string in the answer list will be non-empty.