Question1 As per definition a comment starts with /\* and ends with \*/ . In the body it can

include everything excluding a \* followed by /

1. {SLASH}{STAR}([^\*]\*{STAR}+[^\*/])\*{STAR}+{SLASH}

matches good comment, debatably, /\*\*/ . No comments with body are allowed in this

expression.

2. {SLASH}{STAR}([^\*]|[^/])+{STAR}{SLASH} matches every comment with body. Comments

without body are excluded, which shouldn’t be the case. /\*\*/ is not matched. /\*potato\*/ is

matched but so is /\*pota/\*\*/to and /\*po\*/ta\*/to\*/, which are bad comments. It does not

separate two sequential comments, eg. /\*\*//\*\*/ and counts them as a single comment.

3. {SLASH}{STAR}(.)\*{STAR}{SLASH} The same case as no2 but the difference is that it includes

good comments without body, /\*\*/.

4. {SLASH}{STAR}([^\*]|({STAR}+[^\*/]))\*{STAR}+{SLASH} Matches almost everything properly

but allows bad comments such as /\*pot/\*ato/\* which include /\* multiple times.

5. {SLASH}{STAR}[^(\*/)]\*{STAR}{SLASH} Matches almost everything properly. It does not allow

for bad comments such as /\*pot/\*ato/\* and separates two sequential comments. It does

not, however, allow for comments that include the \* in the body. For example this is a

good comment treated as a bad comment:

/\*This is the header

\* This is some body

\* And I end it like this\*/

Question2.

A possible expression will be: (\#|\/\/)+(.)\*?[\r\n]