CS 536
HW 2
Sparsh Agarwal
9075905142

Answer 1.

1.{SLASH}{STAR}[^(*/)]*{STAR}{SLASH} - This JLex pattern is correct.

  a.The correct comment like, /*h*dv/kzv*/, is passed by the pattern

  b. The incorrect pattern like, /*bjzvd*/*/, is not passed by the pattern.

2. {SLASH}{STAR}(.)*{STAR}{SLASH}- This Jlex pattern is incorrect because it allows */ in the body of the comment.

  a. It allows all the good comments like /*jbzv*/

  b. It also allows the bad comments like, /*zv*/zdv*/

3. {SLASH}{STAR}([^*]*{STAR}+[^*/])*{STAR}+{SLASH}- This Jlex pattern is incorrect because

  a. It allows bad comment like /*v*/**/

  b. It disallows good comments like /**cf*kb/*/

4. {SLASH}{STAR}([^*]|[^/])+{STAR}{SLASH}- This Jlex pattern is incorrect because it does not allow empty body.

  a. It allows bad comments like /*/*/hhhh*/

  b. It disallows good comments like /**/, with empty body

5.{SLASH}{STAR}[^*]*{STAR}+{SLASH}+-This Jlex pattern is incorrect because it allows comments that do not end with */

  a. It allows bad comments like /*xmfbv**////

  b. It disallows good comments with * in body like, /******xfvz**/

6.{SLASH}{STAR}([^*]|({STAR}+[^*/]))*{STAR}+{SLASH}- This Jlex pattern is correct.

  a.The correct comment like, /*h*dv/kzv*/, is passed by the pattern

  b. The incorrect pattern like, /*bjzvd*/*/, is not passed by the pattern.

Answer 2.

Assumptions

  a. The code being scanned is error free

  b. The line numbers are 0-based

```
// User Code
class func{
//fields
static int linenum=0;
static int totalcalls=0;
//methods
static void printcalls(){
System.out.println(totalcalls);
}
Static void printlinenum(){
System.out.println(linenum);
```

```
}
}
%%
// Directives
// Your macro definitions go here
WHITESPACE=   [\040\t]
NEXTLINE=“\n”
POSSPREVCHAR=[,;=(<>&|!?*+-/]
REQSTRING=”myFunc”
POSSNEXTCHAR=”(”

%implements java_cup.runtime.Scanner
%function next_token
%type java_cup.runtime.Symbol

// End of file behavior
%eofval{
func.printcalls();
%eofval}

// Turn on line counting
%line

%%

// Regular expressions rules
({NEXTLINE}|{WHITESPACE}|{POSSPREVCHAR})+{REQSTRING}{WHITESPACE}*{POSSNE
XTCHAR}                {func.linenum=yy.line;
                        func.totalcalls++;
                        func.printlinenum();}
```