

Q1:

$$\{letter\}(\{hyphen\}?( \{ digit \} | \{ letter \} ))*(\#)\{ digit \}+$$

Q2:

$$pattern1 \rightarrow \{SLASH\}\{STAR\}[^{(*)}]^*\{STAR\}\{SLASH\}$$

answer : incorrect

reason : does not allow stars, slashes, left parents and right parents in the comment body

example : `/*****/` is valid comment but not match the pattern
$$pattern2 \rightarrow \{SLASH\}\{STAR\}(.)^*\{STAR\}\{SLASH\}$$

answer : incorrect

reason1 : allow `*/` in the comment bodyexample1: `/*hello */ compiler*/` is not valid but matched

reason2 : does not allow multi lines comment

examplt2: `/* hello``compiler */` is valid but not matched
$$pattern3 \rightarrow \{SLASH\}\{STAR\}([^{*}]^*\{STAR\}+[^{*/}])^*\{STAR\}+\{SLASH\}$$

answer : incorrect

reason : does not allow non stars included comment

example : `/*hello compiler*/` is valid but not matched
$$pattern4 \rightarrow \{SLASH\}\{STAR\}([^{*}]|[^{/}])^*\{STAR\}\{SLASH\}$$

answer : incorrect

reason1 : allow `*/` in the comment bodyexample1 : `/*hello */ compiler*/` is not valid but matched

reason2 : does not allow empty body comment

example2 : `/**/` is valid but not matched
$$pattern5 \rightarrow \{SLASH\}\{STAR\}[^{*}]^*\{STAR\}+\{SLASH\}+$$

answer : incorrect

reason1 : allow unable-to-end comment

example1: `/*hello compiler*/////////` is not valid but matched

reason2 : does not allow stars concat non stars comment

examplt2: `/*hello *****compiler*/` is valid but not matched
$$pattern6 \rightarrow \{SLASH\}\{STAR\}([^{*}]|(\{STAR\}+[^{*/}]))^*\{STAR\}+\{SLASH\}$$

answer : correct

reason : I could not find a example that is valid but not match this pattern or is not valid but match this pattern

Q3:

```
import java_cup.runtime.*; // defines the Symbol class

class FuncCounter {
    static int count=0;
}

%%
WHITESPACE=    [\040\t\n]
CALLMYFUNC=    ("myFunc"){WHITESPACE}*\[.\n]*\

%implements java_cup.runtime.Scanner
%function next_token
%type java_cup.runtime.Symbol
%public
%eofval{
    System.out.println("total number of calls to myFunc is:");
    System.out.println(FuncCounter.count);
    return new Symbol(sym.EOF);
%eofval}

%line

%%
{CALLMYFUNC} {    FuncCounter.cout += 1;
                  System.out.println(yyline+1);
                }
. {;}

```