Given the following grammar, give the First and Follow sets for each non-terminal.

Give the First+ sets for each production

PROG -> STMTS end

STMTS -> STMT STMTS

-> λ

STMT -> EXPR ;

EXPR -> (EXPR) EXPR’

-> TERM

EXPR’ -> PLUSOP (EXPR) EXPR’

-> λ

PLUSOP -> +

-> -

TERM -> [EXPR] TERM’

-> num

TERM’ -> TIMESOP [EXPR] TERM’

-> λ

TIMESOP -> \*

-> /

|  |  |
| --- | --- |
| First | Follow |
| First(PROG) = { end, λ, (, [, num }  First(STMTS) = { λ, (, [, num }  First(STMT) = { (, [, num }  First(EXPR) = { (, [, num }  First(EXPR’) = { λ, +, - }  First(PLUSOP) = { +, - }  First(TERM) = { [, num }  First(TERM’) = { λ, \*, / }  First(TIMESOP) = { \*, / } | Follow(PROG) = {eof}  Follow(STMTS) = { end }  Follow(STMT) = { end }  Follow(EXPR) = { ;, ), ] }  Follow(EXPR’) = { ;, ), ] }  Follow(PLUSOP) = { ( }  Follow(TERM) = { ;, ), ] }  Follow(TERM’) = { ;, ), ] }  Follow(TIMESOP) = { [ } |