**Stack**

[$, prog]

[$, }, stmt\_list, dec\_list, {, ), (, main, type]

[$, }, stmt\_list, dec\_list, {, ), (, main, void]

[$, }, stmt\_list, dec\_list, {, ), (, main]

[$, }, stmt\_list, dec\_list, {, ), (]

[$, }, stmt\_list, dec\_list, {, )]

[$, }, stmt\_list, dec\_list, {]

[$, }, stmt\_list, dec\_list]

[$, }, stmt\_list, ;, id\_list, type]

[$, }, stmt\_list, ;, id\_list, int]

[$, }, stmt\_list, ;, id\_list]

[$, }, stmt\_list, ;, id\_list', id]

[$, }, stmt\_list, ;, id\_list']

[$, }, stmt\_list, ;, id\_list', id, ,]

[$, }, stmt\_list, ;, id\_list', id]

[$, }, stmt\_list, ;, id\_list']

[$, }, stmt\_list, ;, id\_list', id, ,]

[$, }, stmt\_list, ;, id\_list', id]

[$, }, stmt\_list, ;, id\_list']

[$, }, stmt\_list, ;]

[$, }, stmt\_list]

[$, }, stmt\_list', stmt]

[$, }, stmt\_list', read\_stmt]

[$, }, stmt\_list', ;, ), id\_list, (, read]

[$, }, stmt\_list', ;, ), id\_list, (]

[$, }, stmt\_list', ;, ), id\_list]

[$, }, stmt\_list', ;, ), id\_list', id]

[$, }, stmt\_list', ;, ), id\_list']

[$, }, stmt\_list', ;, )]

[$, }, stmt\_list', ;]

[$, }, stmt\_list']

[$, }, stmt\_list', stmt]

[$, }, stmt\_list', write\_stmt]

[$, }, stmt\_list', ;, ), id\_list, (, print]

[$, }, stmt\_list', ;, ), id\_list, (]

[$, }, stmt\_list', ;, ), id\_list]

[$, }, stmt\_list', ;, ), id\_list', id]

[$, }, stmt\_list', ;, ), id\_list']

[$, }, stmt\_list', ;, )]

[$, }, stmt\_list', ;]

[$, }, stmt\_list']

[$, }, stmt\_list', stmt]

[$, }, stmt\_list', if\_stmt]

[$, }, stmt\_list', stmt, else, stmt, ), cndtn, (, if]

[$, }, stmt\_list', stmt, else, stmt, ), cndtn, (]

[$, }, stmt\_list', stmt, else, stmt, ), cndtn]

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp]

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp', term]

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp', term', factor]

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp', term', id]

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp', term']

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop, exp']

[$, }, stmt\_list', stmt, else, stmt, ), exp, relop]

[$, }, stmt\_list', stmt, else, stmt, ), exp, ==]

[$, }, stmt\_list', stmt, else, stmt, ), exp]

[$, }, stmt\_list', stmt, else, stmt, ), exp', term]

[$, }, stmt\_list', stmt, else, stmt, ), exp', term', factor]

[$, }, stmt\_list', stmt, else, stmt, ), exp', term', integer]

[$, }, stmt\_list', stmt, else, stmt, ), exp', term']

[$, }, stmt\_list', stmt, else, stmt, ), exp']

[$, }, stmt\_list', stmt, else, stmt, )]

[$, }, stmt\_list', stmt, else, stmt]

[$, }, stmt\_list', stmt, else, assign]

[$, }, stmt\_list', stmt, else, ;, exp, =, id]

[$, }, stmt\_list', stmt, else, ;, exp, =]

[$, }, stmt\_list', stmt, else, ;, exp]

[$, }, stmt\_list', stmt, else, ;, exp', term]

[$, }, stmt\_list', stmt, else, ;, exp', term', factor]

[$, }, stmt\_list', stmt, else, ;, exp', term', id]

[$, }, stmt\_list', stmt, else, ;, exp', term']

[$, }, stmt\_list', stmt, else, ;, exp', term', factor, \*]

[$, }, stmt\_list', stmt, else, ;, exp', term', factor] d

[$, }, stmt\_list', stmt, else, ;, exp', term', id]

[$, }, stmt\_list', stmt, else, ;, exp', term']

[$, }, stmt\_list', stmt, else, ;, exp']

[$, }, stmt\_list', stmt, else, ;]

[$, }, stmt\_list', stmt, else]

[$, }, stmt\_list', stmt]

[$, }, stmt\_list', assign]

[$, }, stmt\_list', ;, exp, =, id]

[$, }, stmt\_list', ;, exp, =]

[$, }, stmt\_list', ;, exp]

[$, }, stmt\_list', ;, exp', term]

[$, }, stmt\_list', ;, exp', term', factor]

[$, }, stmt\_list', ;, exp', term', id]

[$, }, stmt\_list', ;, exp', term']

[$, }, stmt\_list', ;, exp']

[$, }, stmt\_list', ;]

[$, }, stmt\_list']

[$, }]

[$] $

Correct