

Grammar mC:

program  $\rightarrow$  var\_declaration | function\_declaration

var\_declaration  $\rightarrow$  type list\_id SM

type  $\rightarrow$  INT | FLOAT

list\_id  $\rightarrow$  ID COMA list\_id

| ID

Function\_declaration  $\rightarrow$  type ID parameter\_declaration body\_declaration

parameter\_declaration  $\rightarrow$  LP list\_parameter RP

list\_parameter  $\rightarrow$  not\_empty\_paralist

| empty

not\_empty\_paralist  $\rightarrow$  parameter COMA not\_empty\_paralist

| parameter

parameter  $\rightarrow$  type list\_id

body\_declaration  $\rightarrow$  LB body RB

body  $\rightarrow$  not\_empty\_body

| empty

not\_empty\_body  $\rightarrow$  varDec&stm not\_empty\_body | varDecOrStm

var\_declaration\_or\_statement  $\rightarrow$  vadeclaration | statement

statement  $\rightarrow$  statement SM

| empty

statement  $\rightarrow$  assign | call | return

assign  $\rightarrow$  ID EQUAL exp

call  $\rightarrow$  ID LP list\_expr RP

list\_expr  $\rightarrow$  not\_empty\_exprlist

| empty

not\_empty\_exprlist  $\rightarrow$  exp CM not\_empty\_exprlist

| exp

return  $\rightarrow$  RETURN exp SM

exp  $\rightarrow$  exp1 ADD exp

| exp1

exp1  $\rightarrow$  exp2 SUB exp2

| exp2

exp2  $\rightarrow$  exp2 MUL exp3

| exp2 DIV exp3

| exp3

exp3  $\rightarrow$  operands

operands  $\rightarrow$  INTLIT

| FLOATLIT

| ID

| call

| sub\_exp

sub\_exp  $\rightarrow$  LP exp RP