

program : declarations functions main

;

declarations : ( '#int' id\_list )\*

;

id\_list : ID (',' ID)\*

;

glob\_decl : ( 'global' id\_list )\*

;

functions : function\*

;

function : 'def' ID '(' id\_list ')' ':'

'#{'

declarations

functions

glob\_decl

code\_block

'#}'

;

main : '#def' 'main'

declarations

code\_block

;

code\_block : statement+

;

statement : simple\_statement

| structured\_statement

;

simple\_statement

: assignment\_stat

| print\_stat

| return\_stat

;

structured\_statement

: if\_stat

| while\_stat

;

assignment\_stat : ID '=' ( expression

| 'int' '(' 'input' '(' )

)

;

print\_stat : 'print' '(' expression )'

;

return\_stat : 'return' expression

;

statement\_or\_block

: statement

| '#{ statement+ '#}'

;

if\_stat : 'if' condition ':'

statement\_or\_block

('elif' condition ':'

statement\_or\_block)\*

( 'else' ':'

statement\_or\_block)?

;

while\_stat : 'while' condition ':'

statement\_or\_block

;

expression : optional\_sign term ( add\_oper term )\*

;

term : factor ( mul\_oper factor )\*

;

factor : INTEGER

| '(' expression ')'

| ID idtail

;

idtail : '(' actual\_par\_list ')'

|

;

actual\_par\_list : expression (',' expression)\*

|

;

condition : bool\_term ('or' bool\_term)\*

;

bool\_term : bool\_factor ('and' bool\_factor)\*

;

bool\_factor : expression rel\_op expression

| 'not' expression rel\_op expression

;

optional\_sign : add\_oper

|

;

add\_oper : '+' | '-'

;

mul\_oper : '\*' | '/' | '%'

;

rel\_op : '>' | '<' | '>=' | '<=' | '==' | '!='

