```
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       :  '>' | '<' | '>=' | '<=' | '==' | '!='
```

;