BNF of SOL

program → *func-decl*
func-decl → **func id (** *decl-list-opt* **) :** *domain type-sect-opt var-sect-opt const-sect-opt func-list-opt func-body*
decl-list-opt → *decl-list* | **ε**
decl-list → *decl* **;** *decl-list* | *decl* **;**
decl → *id-list* **:** *domain*
id-list → **id ,** *id-list* | **id**
domain → *atomic-domain* | *struct-domain* | *vector-domain* | **id**
atomic-domain → **char** | **int** | **real** | **string** | **bool**
struct-domain → **struct (** *decl-list* **)**
vector-domain → **vector [intconst] of** *domain*
type-sect-opt → **type** *decl-list* | **ε**
var-sect-opt → **var** *decl-list* | **ε**
const-sect-opt → **const** *const-list* | **ε**
const-list → *const-decl const-list* | *const-decl*
const-decl → *decl* **=** *expr* **;**
func-list-opt → *func-list* | **ε**
func-list → *func-decl func-list* | *func-decl*
func-body → **begin id** *stat-list* **end id**
stat-list → *stat* **;** *stat-list* | *stat* **;**
stat → *assign-stat* | *if-stat* | *while-stat* | *for-stat* | *foreach-stat* | *return-stat* | *read-stat* | *write-stat*
assign-stat → *left-hand-side* **=** *expr*
left-hand-side → **id** | *fielding* | *indexing*
fielding → *left-hand-side* **. id**
indexing → *left-hand-side* **[** *expr* **]**
if-stat → **if** *expr* **then** *stat-list elsif-stat-list-opt else-stat-opt* **endif**
elsif-stat-list-opt → **elsif** *expr* **then** *stat-list elsif-stat-list-opt* | **ε**
else-stat-opt → **else** *stat-list* | **ε**
while-stat → **while** *expr* **do** *stat-list* **endwhile**
for-stat → **for id =** *expr* **to** *expr* **do** *stat-list* **endfor**
foreach-stat → **foreach id in** *expr* **do** *stat-list* **endforeach**
return-stat → **return** *expr*
read-stat → **read** *specifier-opt* **id**
specifier-opt → **[** *expr* **]** | **ε**
write-stat → **write** *specifier-opt expr*
expr → *expr bool-op bool-term* | *bool-term*
bool-op → **and** | **or**
bool-term → *rel-term rel-op rel-term* | *rel-term*
rel-op → == | **!=** | > | >= | < | <= | **in**
rel-term → *rel-term low-bin-op low-term* | *low-term*
low-bin-op → + | **–**
low-term → *low-term high-bin-op factor* | *factor*
high-bin-op → * | /
factor → *unary-op factor* | **(** *expr* **)** | *left-hand-side* | *atomic-const* | *instance-construction* |
func-call | *cond-expr* | *built-in-call* | *dynamic-input*
unary-op → **–** | **not** | *dynamic-output*
atomic-const → **charconst** | **intconst** | **realconst** | **strconst** | **boolconst**
instance-construction → *struct-construction* | *vector-construction*
struct-construction → **struct (** *expr-list* **)**
expr-list → *expr* , *expr-list* | *expr*
vector-construction → **vector (** *expr-list* **)**
func-call → **id (** *expr-list-opt* **)**
expr-list-opt → *expr-list* | **ε**

cond-expr → **if** *expr* **then** *expr elsif-expr-list-opt* **else** *expr* **endif**
elsif-expr-list-opt → **elsif** *expr* **then** *expr elsif-expr-list-opt* | **ε**
built-in-call → *toint-call* | *toreal-call*
toint-call → **toint (** *expr* **)**
toreal-call → **toreal (** *expr* **)**
dynamic-input → **rd** *specifier-opt domain*
dynamic-output → **wr** *specifier-opt*