

Grammatica BNF di NEPAL

program → *type-section* *var-section* *func-section* *run-section*
type-section → **types** *type-decl-list* | **ε**
type-decl-list → *type-decl* *type-decl-list* | *type-decl*
type-decl → *id-list* = *type* ;
id-list → **id** , *id-list* | **id**
type → *atomic-type* | *record-type* | *array-type*
atomic-type → **bool** | **int** | **real** | **string** | **id**
record-type → { *atomic-decl-list* }
atomic-decl-list → *atomic-decl* , *atomic-decl-list* | *atomic-decl*
atomic-decl → **id** : *atomic-type*
array-type → [*array-elem-type*]
array-elem-type → *atomic-type* | *record-type*
var-section → **variables** *var-decl-list* | **ε**
var-decl-list → *var-decl* *var-decl-list* | *var-decl*
var-decl → *id-list* : *type* ;
func-section → **functions** *func-decl-list* | **ε**
func-decl-list → *func-decl* *func-decl-list* | *func-decl*
func-decl → **id** (*opt-formal-decl-list*) : *type* *expr* **end**
opt-formal-decl-list → *formal-decl-list* | **ε**
formal-decl-list → *formal-decl* , *formal-decl-list* | *formal-decl*
formal-decl → **id** : *type*
expr → *expr* *bool-op* *bool-term* | *bool-term*
bool-op → **and** | **or**
bool-term → *comp-term* *comp-op* *comp-term* | *comp-term*
comp-op → **==** | **!=** | **>** | **>=** | **<** | **<=** | **in**
comp-term → *comp-term* *add-op* *term* | *term*
add-op → **+** | **-** | **++**
term → *term* *mul-op* *factor* | *factor*
mul-op → ***** | **/**
factor → *unary-op* *factor* | (*expr*) | *lhs* | *built-in* (*expr*) | *atomic-constant* | *record-constructor* |
 array-constructor | *cond-expr* | *func-call*
unary-op → **-** | **not**
built-in → **int** | **real** | **empty** | **head** | **tail**
atomic-constant → **boolconst** | **intconst** | **realconst** | **strconst**
record-constructor → { *expr-list* }
expr-list → *expr* , *expr-list* | *expr*
cond-expr → **if** *expr* **then** *expr* **else** *expr* **end**
func-call → **id** (*opt-expr-list*)
opt-expr-list → *expr-list* | **ε**
array-constructor → [*opt-expr-list*]
run-section → **run** *stat-list* **end**
stat-list → *stat* ; *stat-list* | *stat* ;
stat → *read-stat* | *write-stat* | *assign-stat* | *if-stat* | *while-stat* | *foreach-stat*
read-stat → **read** **id**
write-stat → **write** *expr*
assign-stat → *lhs* = *expr*
lhs → **id** *opt-fielding* | *indexing*
opt-fielding → . **id** | **ε**
indexing → **id** [*expr*] *opt-fielding*
if-stat → **if** *expr* **then** *stat-list* **else-part** **end**
else-part → **else** *stat-list* | **ε**
while-stat → **while** *expr* **do** *stat-list* **end**
foreach-stat → **foreach** **id** **in** *id* **do** *stat-list* **end**