The grammar for lang allows functions to be defined in inner scopes. For example, the following is legal syntax:

{

int foo()

{

int foo()

{

int foo()

{

return 1;

...

In C, functions can only be declared at global scope (outside all {}). Since lang doesn’t have a global scope, if we wanted an equivalent restriction, we would allow function declarations inside the outer most {} but not inside any inner {}.

For this assignment, I want you to propose grammar changes that would implement this restriction. You only need to provide the portion of the grammar that changes, you don’t need to supply the full grammar.

Example:

{

int foo() // legal

{

int bar(); // not legal

return 1;

}

if (1)

{

int func(); // not legal

}

}

program: prog\_block { $$ = new cProgramNode($1);

yyast\_root = $$;

if (yynerrs == 0)

YYACCEPT;

else

YYABORT;

}

|

block { $$ = new cProgramNode($1);

yyast\_root = $$;

if (yynerrs == 0)

YYACCEPT;

else

YYABORT;

}

prog\_block: open func\_opt stmts close

{ $$ = new cBlockNode($2, $3); }

func\_opt: func\_decls decls

{ $$=$1; $$->Insert($2); }

| func\_decls { $$=$1; }

func\_decls: func\_decls func\_decl

{ $$=$1; $$->Insert($2); }

| func\_decl { $$ = new cDeclsNode($1);

g\_SymbolTable.DecreaseScope();}

decls: decls decl { $$=$1; $$->Insert($2); }

| decl { $$ = new cDeclsNode($1); }

decl: var\_decl ';' { $$ = $1; }

| struct\_decl ';' { $$ = $1; }

| array\_decl ';' { $$ = $1; }

| error ';' {}