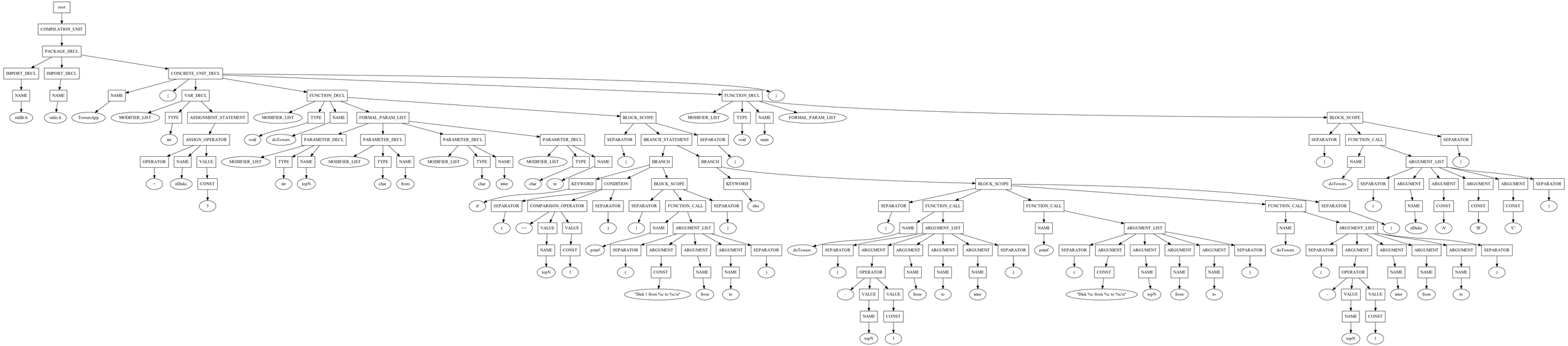
root
COMPILATION_UNIT
PACKAGE_DECL
IMPORT_DECL
NAME
stdlib.h
IMPORT_DECL
NAME
stdio.h
CONCRETE_UNIT_DECL
NAME
TowersApp
{
VAR_DECL
MODIFIER_LIST
TYPE
int
ASSIGNMENT_STATEMENT
ASSIGN_OPERATOR
OPERATOR
=
NAME
nDisks
VALUE
CONST
3
FUNCTION_DECL
MODIFIER_LIST
void
TYPE
doTowers
NAME
FORMAL_PARAM_LIST
PARAMETER_DECL
MODIFIER_LIST
TYPE
int
NAME
topN
PARAMETER_DECL
MODIFIER_LIST
TYPE
char
NAME
from
PARAMETER_DECL
MODIFIER_LIST
TYPE
char
NAME
inter
PARAMETER_DECL
MODIFIER_LIST
TYPE
char
NAME
to
BLOCK_SCOPE
SEPARATOR
{
BRANCH_STATEMENT
BRANCH
KEYWORD
if
CONDITION
SEPARATOR
(
COMPARISON_OPERATOR
==
VALUE
NAME
topN
VALUE
CONST
1
SEPARATOR
)
BLOCK_SCOPE
SEPARATOR
{
FUNCTION_CALL
NAME
printf
ARGUMENT_LIST
SEPARATOR
(
ARGUMENT
CONST
"Disk 1 from %c to %c\n"
ARGUMENT
NAME
from
ARGUMENT
NAME
to
SEPARATOR
)
SEPARATOR
}
BRANCH
KEYWORD
else
BLOCK_SCOPE
SEPARATOR
{
FUNCTION_CALL
NAME
doTowers
ARGUMENT_LIST
SEPARATOR
(
ARGUMENT
OPERATOR
-
VALUE
NAME
topN
VALUE
CONST
1
ARGUMENT
NAME
from
ARGUMENT
NAME
to
ARGUMENT
NAME
inter
SEPARATOR
)
FUNCTION_CALL
NAME
printf
ARGUMENT_LIST
SEPARATOR
(
ARGUMENT
CONST
"Disk %c from %c to %c\n"
ARGUMENT
NAME
topN
ARGUMENT
NAME
from
ARGUMENT
NAME
to
SEPARATOR
)
FUNCTION_CALL
NAME
doTowers
ARGUMENT_LIST
SEPARATOR
(
ARGUMENT
OPERATOR
-
VALUE
NAME
topN
VALUE
CONST
1
ARGUMENT
NAME
inter
ARGUMENT
NAME
from
ARGUMENT
NAME
to
SEPARATOR
)
SEPARATOR
}
BRANCH
SEPARATOR
}
FUNCTION_DECL
MODIFIER_LIST
void
TYPE
main
NAME
FORMAL_PARAM_LIST
BLOCK_SCOPE
SEPARATOR
{
FUNCTION_CALL
NAME
doTowers
ARGUMENT_LIST
SEPARATOR
(
ARGUMENT
NAME
nDisks
ARGUMENT
CONST
'A'
ARGUMENT
CONST
'B'
ARGUMENT
CONST
'C'
SEPARATOR
)
SEPARATOR
}
SEPARATOR
}