root
CompilationUnit
NAME
A
Includes
Include
<
stdio.h
>
TranslationUnit
TranslationUnit
ExternalDeclaration
FunctionDefinition
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
void
Declarator
DirectDeclarator
DirectDeclarator
(
ParameterTypeList
)
insert
ParameterList
ParameterList
,
ParameterDeclaration
ParameterDeclaration
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
char
Declarator
DirectDeclarator
[
]
item
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
int
Declarator
DirectDeclarator
count
CompoundStatement
{
BlockItemList
}
BlockItemList
BlockItemList
BlockItem
BlockItem
Declaration
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
int
InitDeclaratorList
InitDeclarator
Declarator
DirectDeclarator
a
;
Statement
IterationStatement
for
(
ForCondition
)
Expression
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
a
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionConstLiteral
0
;
ForCond
ForExpression
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
<
PrimaryExpressionConstLiteral
10
;
ForExpression
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
++
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
Statement
CompoundStatement
{
BlockItemList
}
BlockItemList
BlockItemList
BlockItem
BlockItem
Statement
ExpressionStatement
Expression
;
IterationStatement
for
(
ForCondition
)
Expression
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionConstLiteral
0
;
ForCond
ForExpression
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
<
PrimaryExpressionConstLiteral
10
;
ForExpression
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
++
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
Statement
CompoundStatement
{
BlockItemList
}
BlockItemList
BlockItem
Statement
ExpressionStatement
Expression
;
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
item
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
a
[
Expression
]
;
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
item
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
b
[
Expression
]
;
ExternalDeclaration
FunctionDefinition
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
void
Declarator
DirectDeclarator
main
(
)
CompoundStatement
{
BlockItemList
}
BlockItemList
BlockItem
BlockItem
Statement
ExpressionStatement
Expression
;
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionConstLiteral
0
;
Declaration
DeclarationSpecifiers
DeclarationSpecifier
TypeSpecifier
char
InitDeclaratorList
InitDeclarator
Declarator
DirectDeclarator
vet
[
]
=
Initializer
{
InitializerList
}
InitializerList
InitializerList
,
Initializer
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionConstLiteral
'Y'
'V'
'T'
'A'
'L'
'I'
'D'
'a'
'l'
'i'
'v'
'e'
'\0'
;
Statement
ExpressionStatement
Expression
;
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
item
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
i
[
Expression
]
;
AssignmentExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
item
AssignmentOperator
=
AssignmentExpression
ConditionalExpression
LogicalOrExpression
LogicalAndExpression
InclusiveOrExpression
ExclusiveOrExpression
AndExpression
EqualityExpression
RelationalExpression
ShiftExpression
AdditiveExpression
MultiplicativeExpression
CastExpression
UnaryExpression
PostfixExpression
PrimaryExpression
PrimaryExpressionVar
b
[
Expression
]
;
Statement
ExpressionStatement
Expression