**C--文法：**

1. program -> compUnit //程序 -> 编译单元

2. compUnit -> decl compUnit //编译单元 -> 声明 编译单元

3. compUnit -> funcDef compUnit //编译单元 -> 函数定义 编译单元

compUnit -> struDef compUnit //编译单元 -> 结构体定义 编译单元

4. compUnit -> $ //编译单元 -> 空

5. decl -> constDecl //声明 -> 常量声明

6. decl -> varDecl //声明 -> 变量声明

decl -> struDecl //声明 -> 结构体声明

7. constDecl -> const bType constDef argConst ; //常量声明 -> const 数据类型 常量定义 可选常量 ;

8. argConst -> , constDef argConst //可选常量 -> , 常量定义 可选常量

9. argConst -> $ //可选常量 -> 空

10. constDef -> Ident = constInitVal //常量定义 -> 标识符 = 常量初值

11. constInitVal -> constExp //常量初值 -> 常量表达式

12. varDecl -> bType varDef argVarDecl ; //变量声明 -> 数据类型 变量定义 可选变量声明 ;

13. argVarDecl -> , varDef argVarDecl //可选变量声明 -> , 变量定义 可选变量声明

14. argVarDecl -> $ //可选变量声明 -> 空

15. varDef -> Ident argVarDef //变量定义 -> 标识符 可选变量定义

16. argVarDef -> = initVal //可选变量定义 -> = 变量初值

17. argVarDef -> $ //可选变量定义 -> 空

18. initVal -> exp //变量初值 -> 表达式

struDecl -> struct struType Ident //结构体声明 -> struct 结构体类型 标识符

struType -> Ident //结构体类型 -> 标识符

struDef -> struct Ident { struElem } struVar ;//结构体定义 -> struct 标识符 { 结构体成员 } 附带声明;

struVar -> Ident argStruVar //附带声明 -> 标识符 可选附带声明

argStruVar -> , Ident argStruVar //可选附带声明 -> , 标识符 可选附带声明

argStruVar -> $ //可选附带声明 -> 空

struElem -> elemDe argStruElem //结构体成员 -> 成员定义或声明 可选成员

argStruElem -> elemDe argStruElem //可选成员 -> 成员定义或声明 可选成员

argStruElem -> $ //可选成员 -> 空

elemDe -> decl ; //成员定义或声明 -> 声明;

elemDe -> funcDef //成员定义或声明 -> 函数定义

19. bType -> int //数据类型 -> int

20. funcDef -> funcType Ident ( funcFParams ) block //函数定义 -> 函数类型 标识符 ( 函数形参 ) 语块

21. funcType -> void //函数类型 -> void

22. funcFParams -> funcFParam argFunctionF //函数形参表 -> 函数形参 可选函数形参

23. funcFParams -> $ //函数形参表 -> 空

24. argFunctionF -> , funcFParam argFunctionF //可选函数形参 -> , 函数形参，可选函数形参

25. argFunctionF -> $ //可选函数形参 -> 空

26. funcFParam -> bType Ident //函数形参 -> 数据类型 标识符

funcFParam -> struct struType Ident //函数形参 -> struct 结构体类型 标识符

27. block -> { blockItem } //语块 -> { 语块项 }

28. blockItem -> decl blockItem //语块项 -> 声明 语块项

29. blockItem -> stmt blockItem //语块项 -> 语句 语块项

30. blockItem -> $ //语块项 -> 空

31. stmt -> exp ; //语句 -> 表达式 ;

32. stmt -> ; //语句 -> ;

33. stmt -> block //语句 -> 语块

34. stmt -> return argExp ; //语句 -> return 可选表达式 ;

35. argExp -> $ //可选表达式 -> 空

36. argExp -> exp //可选表达式 -> 表达式

37. exp -> assignExp //表达式 -> 赋值表达式

38. lVal -> Ident //左值表达式 -> 标识符 后附左值表达式解释

39. primaryExp -> ( exp ) //基本表达式 -> ( 表达式 )

40. primaryExp -> number //基本表达式 -> 数

41. number -> INT //数 -> 整数

42. unaryOp -> + //一元运算符 -> +

43. unaryOp -> - //一元运算符 -> -

44. unaryOp -> ! //一元运算符 -> !

45. unaryExp -> unaryOp unaryExp //一元表达式 -> 一元运算符 一元表达式

46. unaryExp -> Ident callFunc //一元表达式 -> 标识符 函数调用

47. callFunc -> ( funcRParams ) //函数调用 -> ( 函数实参表 )

48. callFunc -> $ //函数调用 -> 空

49. unaryExp -> primaryExp //一元表达式 -> 基本表达式

50. funcRParams -> funcRParam argFunctionR //函数实参表 -> 函数实参 可选函数实参

51. funcRParams -> $ //函数实参表 -> 空

52. argFunctionR -> , funcRParam argFunctionR //可选函数实参 -> , 函数实参 可选函数实参

53. argFunctionR -> $ //可选函数实参 -> 空

54. funcRParam -> exp //函数实参 -> 表达式

55. mulExp -> unaryExp mulExpAtom //乘除表达式 -> 一元表达式 乘除表达式原子

56. mulExpAtom -> \* unaryExp mulExpAtom //乘除表达式原子 -> \* 一元表达式 乘除表达式原子

57. mulExpAtom -> / unaryExp mulExpAtom //乘除表达式原子 -> / 一元表达式 乘除表达式原子

58. mulExpAtom -> % unaryExp mulExpAtom //乘除表达式原子 -> % 一元表达式 乘除表达式原子

59. mulExpAtom -> $ //乘除表达式原子 -> 空

60. addExp -> mulExp addExpAtom //加减表达式 -> 乘除表达式 加减表达式原子

61. addExpAtom -> + mulExp addExpAtom //加减表达式原子 -> + 乘除表达式 加减表达式原子

62. addExpAtom -> - mulExp addExpAtom //加减表达式原子 -> - 乘除表达式 加减表达式原子

63. addExpAtom -> $ //加减表达式原子 -> 空

64. relExp -> addExp relExpAtom //关系表达式 -> 加减表达式 关系表达式原子

65. relExpAtom -> < addExp relExpAtom //关系表达式原子 -> < 加减表达式 关系表达式原子

66. relExpAtom -> > addExp relExpAtom //关系表达式原子 -> > 加减表达式 关系表达式原子

67. relExpAtom -> <= addExp relExpAtom //关系表达式原子 -> <= 加减表达式 关系表达式原子

68. relExpAtom -> >= addExp relExpAtom //关系表达式原子 -> >= 加减表达式 关系表达式原子

69. relExpAtom -> $ //关系表达式原子 -> 空

70. eqExp -> relExp eqExpAtom //相等性表达式 -> 关系表达式 相等性表达式原子

71. eqExpAtom -> == relExp eqExpAtom //相等性表达式原子 -> == 关系表达式 相等性表达式原子

72. eqExpAtom -> != relExp eqExpAtom //相等性表达式原子 -> != 关系表达式 相等性表达式原子

73. eqExpAtom -> $ //相等性表达式原子 -> 空

74. assignExp -> eqExp assignExpAtom //赋值表达式 -> 相等性表达式 赋值表达式原子

75. assignExpAtom -> = eqExp assignExpAtom //赋值表达式原子 -> = 相等性表达式 赋值表达式原子

76. assignExpAtom -> $ //赋值表达式原子 -> 空

77. constExp -> assignExp //常量表达式 -> 赋值表达式

