# mission\_2.1.5.2

## 关于声明的问答

* 1. 某一行代码”int a;”叫做声明，声明（不是）语句。
  2. 声明”int a;”后面一部分叫做标识符，请问1a是合理的标识符吗？为什么？ 不是，标识符第一位不能是数字 。
  3. 实际上，现在的C标准中，合理的声明说明符还有一条long double。按照C语言的惯例，你可以推断long double类型的变量最少会占有 8 字节的内存。
  4. 声明说明符和类型说明符之间的关系是什么？ 声明说明符是有多个类型说明符组成的 。
  5. 声明”char a;”说明，内存中预留了一个大小为 1 字节，数据类型为 字符 ， 标识符 为a的变量。

## 关于语法正式定义的问答

* 1. 语法的正式定义建立在非常美妙紧凑的书写方法上，斜体就表示语法成分，我们不断对斜体进行类似于递归的定义，语法就可以按照这些规则发展成一个“语法树”。

*declaration-specifiers：*

*type-specifiers declaration-specifiersOPT*

1. 这个语句意味着，假设说目前我有一个语法成分是*declaration-specifiers*，那么我可以把它展开成*type-specifiers declaration-specifiers*。（选择可选成分）

2. 对于1中新的展开式中，后面又出现了一个*declaration-specifiers*，它还可以按照定义展开。所以整个式子还可以展开成*type-specifiers type-specifiers declaration-specifiers*。（选择可选成分）

3. 对于2中新的展开式中，后面还有一个*declaration-specifiers*，这一次我决定将他展开，且不选择语句定义后面的可选成分，于是整个式子就变成了 *type-specifiers type-specifiers type-specifiers* 。

4. 这说明*declaration-specifiers*和*type-specifiers*之间的关系是： *declaration-specifiers由若干个type-specifiers组成* 。

* 1. 我们描述*element1*，*element2*与*element3*有如下的关系：*element1*是由若干个*element2*和*element3*的混合组合而成。

需要注意的是：定义语句中出现多行，则表示多行中选择一个，例如说

*element1：*

*element2*

*elemen3*

这意味着*element1*可能是*element2*或者*element3*。

请写出题干中*element1*的真实定义，请善用OPT可选这个语法。

*element1：*

*element2 element1OPT*

*element3 element1OPT*

2.2 现在我们试着给标识符*identifier*书写一份正式的语法定义：

标识符有两个限制：

1. 标识符可以由数字和大小写字母以及下划线组成，但是标识符的第一位不能是数字。

2. 标识符不能和关键字相同。

现在已知定义：

nondigit: one of  
**\_ a b c d e f g h i j k l mn o p q r s t u v w x y z  
    A B C D E F G H I J K L MN O P Q R S T U V W X Y Z**

digit: one of  
**0 1 2 3 4 5 6 7 8 9**

(这里的one of 是为了让排版更好看，否则为了展示多选一的关系这个定义非要上下并列书写，内容就太长了！)

书写定义：

*identifier*:

*nondigit*

*identifier nondigit*

*identifier* *digit*

* 1. 一个按照语法规定展开的内容，一定就是正确的C程序的内容吗？为什么？ 不一定正确，例如说 unsigned int long char 或者 int unsigned 虽然都是语法上合理的东西，但是这样的声明标识符不是C程序接受的声明 。