**C语言常量**

1. 常量：（不可以定义结构型常量） 常量包括字面常量，符号常量，契约型常量，枚举常量这几种类型。
2. 字面常量包括：整形常量，字符型常量，字符串常量。注意：不存在数组常量，结构体常量等结构型的字面常量。但是存在结构型的符号常量

3.符号常量：（可以定义结构型常量） 用#define和const定义的常量！这两种常量之间的区别：

* #define定义的常量，除了字符串字面常量外都不占内存，所以无法取常量的地址，仅仅是宏替换而已eg：#define NAME “pang dong” 本质是字符串字面常量，会占用“静态存储区” #define MAX 256 本质是整形的字面常量，不会分配内存
* const定义的常量要分情况考虑： 你可以去一个const符号常量的地址：对于基本数据类型的const常量，编译器会重新在内存中创建一个他的一个拷贝，（真正的基本数据类型的const常量会被编译器放到符号表中而不分配内存空间）你通过其地址访问到的就是这个拷贝而不是原始的符号常量；而对于构造类型的const常量，实际上他是编译时不允许修改的常量，是占内存空间的，因此如果你能绕过编译器的静态类型安全检查机制，就可以在运行时修改其内存单元。从理论上讲，只要握有一个对象（针对C++中的结构类型常量）的指针，你就可以设法绕过编译器修改它的内存。详见《林锐-第三版》P86

4. 契约常量

    未使用关键字const但是被看做一个const对象的

    void ReadValue(cons tint & num){ cout << num ;}

int main(){ int n = 999; ReadValue(n);}

5.枚举常量

    C/C++的构造类型enum实际上是用来定义一些相关常量的集合。并且规定枚举常量的值是可以扩展的，并非受限一般的整数类型。

    enum Gigntic{ SMALL = 10, MID = 100, BIG = 1000};底层如何实现根据C/C++提供山的不同实现而有所不同。

6.使用符号常量的好处？如果不使用符号常量，而是直接在程序中填写数字或字符串，将会有哪些麻烦呢？

* 程序的可读性（可理解性）变差。程序员自己会忘记那些数字或字符串是什么意思，用户则更加不知它们从何处来、表示什么。
* 在程序的很多地方输入同样的数字或字符串，难保不发生书写错误。
* 如果要修改数字或字符串，则会在很多地方改动，既麻烦又容易出错。

7.const与#define的比较C++语言可以用const来定义常量，也可以用#define来定义常量。但是前者比后者有更多的优点：

* const常量有数据类型，而宏常量没有数据类型。编译器可以对前者进行类型安全检查。而对后者只进行字符替换，没有类型安全检查，并且在字符替换可能会产生意料不到的错误（边际效应）。
* 有些集成化的调试工具可以对const常量进行调试，但是不能对宏常量进行调试。【规则】在C++程序中只使用const常量而不使用宏常量，即const常量完全取代宏常量。