在C++的Template中很多地方都用到了typename与class这两个关键字，有时候这两者可以替换，那么这两个关键字是否完全一样呢？

　　事实上class用于定义类，在模板引入c++后，最初定义模板的方法为:template<class T>,这里class关键字表明T是一个类型，后来为了避免class在这两个地方的使用可能给人带来混淆，所以引入了typename这个关键字，它的作用同class一样表明后面的符号为一个类型，这样在定义模板的时候可以使用下面的方式了:

template<typename T>.在模板定义语法中关键字class与typename的作用完全一样。

我们已经学过重载(Overloading)，对重载函数而言,C++的检查机制能通过函数参数的不同及所属类的不同。正确的调用重载函数。例如，为求两个数的最大值，我们定义MAX()函数需要对不同的数据类型分别定义不同重载(Overload)版本。

//函数1.

int max(int x,int y)

{ return(x>y)?x:y ; }

//函数2.

float max( float x,float y)

{ return (x>y)? x:y ; }

//函数3.

double max(double x,double y)

{ return (x>y)? x:y ; }

但如果在主函数中，我们分别定义了 char a,b; 那么在执行max(a,b);时 程序就会出错，因为我们没有定义char类型的重载版本。

现在，我们再重新审视上述的max()函数，它们都具有同样的功能，即求两个数的最大值，能否只写一套代码解决这个问题呢？这样就会避免因重载函数定义不 全面而带来的调用错误。为解决上述问题C++引入模板机制，模板定义：模板就是实现代码重用机制的一种工具，它可以实现类型参数化，即把类型定义为参数，从而实现了真正的代码可重用性。模版可以分为两类，一个是函数模版，另外一个是类模版。

template <class T>

T min(T x,T y)

{

return(x<y)?x:y;

}