

```
第1行    using namespace std;
第2行    //包装一元函数指针，使之成为具有仿函数一样能力
第3行    //定义一个指向一元函数的类
第4行    //Result:返回值类型，Arg是参数
第5行    template<typename Arg,typename Result>
第6行    class Pointer_to_unary_function
第7行    {
第8行    public unary_function<Arg , Result>
第9行    protected:
第10行    Result (*ptr)(Arg);
第11行    public:
第12行    Pointer_to_unary_function() {}//空构造函数
第13行    explicit Pointer_to_unary_function(Result (*x)(Arg)):ptr(x) {}//构造函数
第14行    Result operator()(Arg x)const{return ptr(x);}
第15行    };
第16行    //为规避干扰，将ptr_fun改为ptrFun
第17行    template<typename Arg,class Result>
第18行    inline Pointer_to_unary_function<Arg,Result> ptrFun(Result (*x)(Arg)) {
第19行        return Pointer_to_unary_function<Arg,Result>(x);
第20行    };
第21行
第22行    //包装二元函数指针，使之成为具有仿函数一样能力
第23行    //定义一个指二元函数的类
第24行    //Result:返回值类型，Arg1、Arg2是参数
第25行    template<typename Arg1,typename Arg2,typename Result>
第26行    class Pointer_to_binary_function:public binary_function<Arg1,Arg2,Result>
第27行    {
第28行    protected:
第29行    Result (*ptr)(Arg1,Arg2);
第30行    public:
第31行    Pointer_to_binary_function() {}//空构造函数
第32行    explicit Pointer_to_binary_function(Result (*x)(Arg1,Arg2)):ptr(x) {}//构造函数
第33行    Result operator()(Arg1 x,Arg2 y)const{return ptr(x,y);}
第34行    };
第35行    //为规避干扰，将ptr_fun改为ptrFun
第36行    template<typename Arg1,typename Arg2,class Result>
第37行    inline Pointer_to_binary_function<Arg1,Arg2,Result> ptrFun(Result (*x)(Arg1,Arg2)) {
第38行        return Pointer_to_binary_function<Arg1,Arg2,Result>(x);
第39行    };
```