mem\_fn()：调用成员函数

例：程序stl\_test61

class Person {

public:

Person(const *string*& name) : name\_(name) {

}

void Print() const {

*cout* << name\_ << *endl*;

}

private:

*string* name\_;

};

int main(int argc, char\* argv[])

{

// mem\_fn()

*vector*<Person> coll = { Person("Tick"), Person("Trick"), Person("Track") };

*for\_each*(coll.*begin*(), coll.*end*(), *mem\_fn*(&Person::Print));

return 0;

}

输出为：

Tick

Trick

Track

function adapter搭配用户自定义的function object

例：程序stl\_test62

template <typename T, typename V>

struct fopow

{

T operator() (T base, V exp) const

{

return std::*pow*(base, exp);

}

};

*vector*<int> coll = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

// 用coll中的元素作为幂，底数为3

*transform*(coll.*begin*(), coll.*end*(), // source

*ostream\_iterator*<float>(*cout*, " "), // destination

*bind*(fopow<float, int>(), 3, *\_1*)); // operation

*cout* << *endl*;

// 用3做幂，底数为coll的元素

*transform*(coll.*begin*(), coll.*end*(),

*ostream\_iterator*<float>(*cout*, " "),

*bind*(fopow<float, int>(), *\_1*, 3));

*cout* << *endl*;

输出为：

3 9 27 81 243 729 2187 6561 19683

1 8 27 64 125 216 343 512 729