反向适配器：

例：程序stl\_test56

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

// 元素5所在的位置

*vector*<int>::*const\_iterator* pos = *find*(coll.*cbegin*(), coll.*cend*(), 5);

*cout* << "pos: " << \*pos << *endl*;

// 反向迭代器

*vector*<int>::*const\_reverse\_iterator* rpos(pos);

*cout* << "rpos: " << \*rpos << *endl*;

*cout* << \*coll.*rbegin*() << *endl*; // 9

// 内存错误

//cout << \*coll.rend() << endl;

输出为：

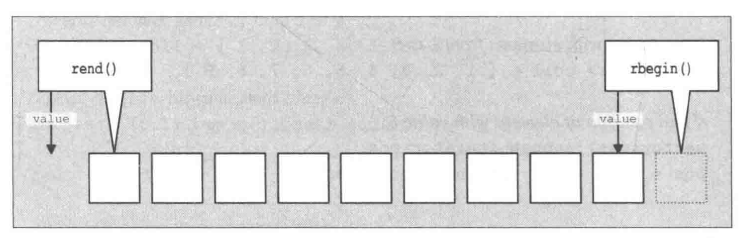
pos: 5

rpos: 4

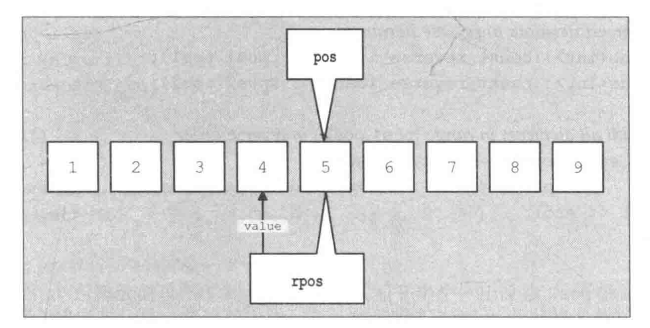
9

从输出可以看出：同一个位置的正向迭代器和反向迭代器的值不同。

Reverse迭代器所定义的逻辑区间为[rbegin, rend)，但实际的物理区间不包括起点，却包含了终点(begin, end]，其逻辑上指向的元素位置就和物理上值的元素位置不同。



所以\*rbegin有值，而\*rend会报内存位置



正向迭代器其物理位置与逻辑位置一致，反向迭代器其物理位置与逻辑位置不一致。