stack和queue的底层模拟实现是deque来实现的

```
//stack<int> st;
template<class _Ty, class _Container = deque<_Ty> >
class stack
{
public:
	stack()	: c()
	{}
	void push(value_type&& _Val)
	{
		c.push_back(val);
	}

	bool empty() const
	{	// test if stack is empty
		return (c.empty());
	}

	size_type size() const
	{	// test length of stack
		return (c.size());
	}

	reference top()
	{	// return last element of mutable stack
		return (c.back());
	}

	const_reference top() const
	{	// return last element of nonmutable stack
		return (c.back());
	}

	void pop()
	{	// erase last element
		c.pop_back();
	}
protected:
	_Container c;	// the underlying container
};


void main()
{
	stack<int> st;
```

```
    for(int i=1; i<=5; ++i)
        st.push(i);
    cout<<"size = "<<st.size()<<endl;

    int value;
    while(!st.empty())
    {
        value = st.top();
        st.pop();
        cout<<value<<" ";
    }
    cout<<endl;
}
```

stack和queue都是有默认容器适配器参数是vector

```
void main()
{
    queue<int, vector<int> >q;
    q.push(1);
    //q.pop();

}
```