```
/*This code shows examples of container-specific iterators and how to handle them*/

#include <iostream>
#include <vector>
#include <list>
#include <iterator> //Enable the code to use "iterator"
using namespace std;

void main() {

	vector<int> vec{ 0, 1, 2, 3 }; //random access only

	list<int> lis{ 0, 1, 2, 3 }; //bidirectional access only

	int* pvec = &vec[0]; // Raw pointer

	cout << *(++pvec) << endl;

	//int* plis = &lis[0]; // This won't work, do you know why?

	// Defining a container-specific iterator:

	list<int>::iterator plis = lis.begin();

	// or simply
	//auto plis = lis.begin();

	cout << *(++plis) << endl;

	//plis += 2; // This won't work, do you know why?

	system("pause");
}
```