```
#include <vector>
#include <iostream>

using namespace std;


…
{
vector<int> ti;

	ti.push_back(1);
//dodanie pojedynczego elementu na koniec wektora,
//pojemność zaalokowanej pamięci może się podwoić
	ti.push_back(2);
	for (int i = 0; i < ti.size(); i++) //size() zwraca liczbę elementów w wektorze
		cout << ti[i] << " "; //zawartość 1,2; ti[..] dostęp do pojedynczego elementu
wektora
	cout << endl;

	ti.pop_back(); // usunięcie ostatniego elementu z wektora
	for (int i = 0; i < ti.size(); i++)
		cout << ti[i] << " ";  //zawartość ti to 1
	cout << endl;

	ti.push_back(2);
	ti.push_back(3);
	ti.erase(ti.begin() + 1); //usunięcie drugiego elementu z wektora
	for (int i = 0; i < ti.size(); i++)
		cout << ti[i] << " "; //zawartość ti to 1,3
	cout << endl;

	cout << ti.size()<< " " << ti.capacity() << endl;
//zwraca liczbę elementów wektora oraz aktualną pojemność wektora
//wyrażoną w liczbie elementów
//pojemność jest zawsze większa lub równa liczbie elementów

	ti.reserve(10);
//zwiększenie zarezerwowanej pamięci, tak aby pomieściła zadaną liczbę elementów
	cout << ti.size() << " " << ti.capacity() << endl; //zwraca 2,10

	cout << ti.front() << " " << ti.back() << endl;
//zwraca pierwszy i ostatni element wektora, czyli 1 i 3

	ti.clear(); // usuwa wszystkie elementy ze zbioru
} //tutaj ten wektor jest automatycznie usuwany

////////////////////////////////////////
#include <set>
#include <iostream>

using namespace std;

...
	set<int> zb; //zbiór, w którym elementy o tej zamej wartości nie mogą się powtarzać
	// multiset<> zbiór w którym elementy o tej samej wartości mogą się powtarzać

	zb.insert(1); zb.insert(2); zb.insert(3);
	cout << "wartosc 1 wystepuje " << zb.count(1) << " razy\n";

	zb.erase(1); //usunięcie elementu o wartości 1
	if (zb.find(1) == zb.end() )
		cout << "nie wystepuje element o wartosci 3 \n";

	zb.insert(1); zb.insert(2); zb.insert(3); zb.insert(4);
```

```
for (set<int>::iterator it = zb.begin(); it != zb.end(); it++)
       cout << *it << " ";  //zwraca elementy 1,2,3,4
cout << endl;
```