vector<T>		
	Δήλωση/δημιουργία	vector<int> v; (stack) vector<int> * pv = new vector<int>(); (heap)
	Εισαγωγή στοιχείου στο τέλος	v.push_back(5);
	Διαγραφή στοιχείου από το τέλος	v.pop_back();
	Εισαγωγή στοιχείου σε τυχαία θέση	v.insert(v.begin()+n, value)
	Διαγραφή στοιχείου σε τυχαία θέση	v.erase(v.begin()+n)
ΠΡΟΣΠΕΛΑΣΗ	**Προσπέλαση στοιχείου στη θέση n**	int value = v[n]; int value = v.at(n);
	Πρώτο στοιχείο	Int value = v.front();
	Τελευταίο στοιχείο	Int value = v.back();
	Πλήθος στοιχείων	int size = v.size();

	Διάσχιση	for (int i=0;i<v.size();i++) {
		// do sth to v[i]
		}
		
		for (vector<int>::iterator it = v.begin() ; it != v.end() ; ++it) {
		// do sth to *it
		}

	Διαγραφή όλων των στοιχείων	v.clear();

map<K,V> unordered_map<K,V> k:unique	
Δήλωση	map<string, int> m; map<string, int> * pm = new map<string, int>();

Εισαγωγή ζεύγους	m["bob"] = 10;
	m["eva"] = 10;
	m[key] = value;

Διαγραφή στοιχείου	m.erase("bob");
Αλλαγή τιμής στο στοιχείο με κλειδί bob	m["bob"] = 11; m["bob"]++;
Προσπέλαση στο στοιχείο με κλειδί bob	cout << m["bob"]
Πλήθος στοιχείων	int size = m.size();
Διαγραφή όλων των στοιχείων	m.clear();

Διάσχιση	for (map<string, int>::iterator it=m.begin() ; it != m.end() ; ++it) {
	// do sth to it->first (key)
	// do sth to it->second (value)
	}

Εύρεση στοιχείου με κλειδί bob	map<string, int>::iterator it = m.find("bob");
	
	if (it != m.end()) {
	// bob exists map ...
	} else {
	// bob does not exist in map
	}