

# Εργαστήριο 5ο

## Containers – Iterators - Algorithms

### Άσκηση 1 (vectors)

Να δημιουργήσετε το παρακάτω πρόγραμμα Lab5E1 που παρουσιάζει την δημιουργία και τις βασικές χρήσεις των vectors.

```
#include <iostream>
#include<vector>
using namespace std;

int main()
{
    vector<int> ivec{ 0,1,2,3,4,5 };

    ivec.push_back(44);
    ivec.insert(ivec.begin() + 2, 33);

    for (auto x : ivec) {
        cout << "Elem:" << x << endl;
    }
    cout << "Size=" << ivec.size() << endl;

    ivec.erase(ivec.end() - 1);
    cout << "After erase, Size=" << ivec.size() << endl;

    for (auto& x : ivec)
        x = 0;
    cout << "After x=0, Size=" << ivec.size() << endl;

    ivec.clear();
    cout << "After clear, Size=" << ivec.size() << endl;
}
```

Απαντήστε στις παρακάτω ερωτήσεις

1. Πως ορίζουμε ένα vector; Πρέπει να κάνουμε include κάποια βιβλιοθήκη;
2. Με ποιόν τρόπο μπορούμε να αρχικοποιούμε ένα vector;
3. Μπορούμε να έχουμε ένα vector που να περιέχει διαφορετικού τύπου στοιχεία;
4. Με ποιόν τρόπο μπορούμε να προσθέτουμε στοιχεία σε έναν vector; Αυτά τα στοιχεία που τοποθετούνται;
5. Με ποιον τρόπο μπορούμε να τοποθετήσουμε ένα στοιχείο σε κάποια συγκεκριμένη θέση;
6. Πως διαγράφουμε ένα στοιχείο σε μία συγκεκριμένη θέση;
7. Με ποιον τρόπο μπορούμε να έχουμε πρόσβαση στα στοιχεία ενός vector; Τι γίνεται αν θέλουμε να μεταβάλουμε τα στοιχεία αυτά;
8. Πως είμαστε σίγουροι ότι δεν ξεπερνάμε τα «όρια» ενός vector
9. Πως διαγράφουμε όλα τα στοιχεία ενός vector

Να εκτελέσετε τις παρακάτω αλλαγές (μία κάθε φορά)

1. Αντικαταστήστε την εντολή «vec.push\_bask(8)» με την «v[3]=8;». Δουλεύει το πρόγραμμα; Γιατί;
2. Αντικαταστήστε την range for με μία κανονική for

## Άσκηση 2 (iterators - algorithms)

Να δημιουργήσετε το παρακάτω πρόγραμμα Lab5E2 που παρουσιάζει την δημιουργία και τις βασικές χρήσεις των iterators στους αλγόριθμους.

```
#include<iostream>
#include<vector>
#include<string>
#include<algorithm>
using namespace std;

int main() {
    vector<string> text;

    auto ins = back_inserter(text);
    string word;

    cout << "Give words (ctrl-z to end):\n";
    while (cin >> word)
        *ins = word;

    cout << "All words\n";
    for (auto x = text.begin(); x != text.end(); ++x)
        cout << "-" << *x << endl;

    sort(text.begin(), text.end());

    cout << "After sort\n";
    for (auto x = text.begin(); x != text.end(); ++x)
        cout << "-" << *x << endl;

    auto res = unique(text.begin(), text.end());

    vector<string> newtext(text.begin(), res); // or copy
    cout << "After unique\n";
    for (auto x = newtext.begin(); x != newtext.end(); ++x)
        cout << "-" << *x << endl;
}
```

Απαντήστε στις παρακάτω ερωτήσεις

1. Πως δημιουργείται ένας iterator σε ένα container; Ποια είναι η διαφορά ενός const από ένα κανονικό iterator;
2. Πως δημιουργείται ένας inserter iterator (back); Ποια είναι η διαφορά του από τους κανονικούς iterators;
3. Ποιες είναι οι πράξεις που μπορούμε να κάνουμε σε έναν iterator;
4. Πως μπορούμε να έχουμε πρόσβαση στα στοιχεία ενός container μέσω ενός operator;
5. Τι σημαίνει η διαφορά μεταξύ δύο iterator;
6. Τι είναι ένα iterator range;
7. Να περιγράψετε δύο γενικούς αλγόριθμους και την λειτουργία τους

Να εκτελέσετε τις παρακάτω αλλαγές (μία κάθε φορά)

1. Να μετατρέψετε το πρόγραμμα έτσι ώστε να εμφανίζει μόνο τις διπλές λέξεις

## Άσκηση 3 (maps)

Να δημιουργήσετε το παρακάτω πρόγραμμα Lab5E3 που παρουσιάζει την δημιουργία και τις βασικές χρήσεις των maps

```
#include<iostream>
#include<map>
#include<string>
using namespace std;

int main() {
    map<int, string> words{ {1,"ena"},{2,"dya"} };

    words[1] = "Ena";
    words[3] = "tria";

    auto res = words.insert({ 4,"tessera" });

    cout << "Egine eisagosi:" << res.second << endl;
    cout << "Stoixeiio poy mpike (key):" << res.first->first << endl;
    cout << "Stoixeiio poy mpike (value):" << res.first->second << endl;

    words.erase(3);

    for (auto x : words)
        cout << x.first << "->" << x.second << endl;
}
```

Απαντήστε στις παρακάτω ερωτήσεις

1. Πως δημιουργείται ένα map;
2. Πως έχουμε πρόσβαση στα στοιχεία ενός map;
3. Πως μπορούμε να εισάγουμε καινούργια στοιχεία σε ένα map;
4. Μπορεί δύο στοιχεία ενός map να έχουν το ίδιο κλειδί;
5. Τι γίνεται αν εισάγουμε ένα στοιχείο με ένα κλειδί που υπάρχει ήδη;
6. Πως ελέγχουμε αν ένα κλειδί υπάρχει ήδη σε ένα map;
7. Πως διαγράφουμε στοιχεία ενός map;
8. Ποιος είναι ο τύπος των στοιχείων ενός map;

## Άσκηση 4

Να δημιουργήσετε ένα πρόγραμμα που θα συνενώνει δύο vectors που περιέχουν στοιχεία τύπου int. Το τελικό vector δεν πρέπει να είναι ταξινομημένο με φθίνουσα σειρά και να μην έχει διπλά στοιχεία.

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;

int main() {
    vector<int> vec1{ 2,5,13,7,9,11,56 };
    vector<int> vec2{ 3,5,8,4,11,3,56,7 };
    vector<int> tmpvec;
    vector<int> finvec;

    // concat
    tmpvec = vec1;
    auto bit = back_inserter(tmpvec);
```

```
copy(vec2.begin(), vec2.end(), bit);

//sort
sort(tmpvec.begin(), tmpvec.end());

//unique
auto res = unique(tmpvec.begin(), tmpvec.end());
auto bit2 = back_inserter(finvec);
copy(tmpvec.begin(), res, bit2);

// reverse
vector<int> rvec;
auto cur = finvec.end();
auto bit3 = back_inserter(rvec);

do {
    cur--;
    *bit3 = *cur;
} while (cur != finvec.begin());

for (auto x : rvec)
    cout << x << endl;
}
```

## Άσκηση 5

Να δημιουργήσετε ένα πρόγραμμα που θα διαβάζει τις λέξεις ενός αρχείου. Θα δημιουργεί δύο vectors ο πρώτος θα είναι οι μοναδικές λέξεις που περιέχει το αρχείο και ο δεύτερος ο αριθμός εμφάνισης της αντίστοιχης λέξης.

## Άσκηση 6

Να δημιουργήσετε το παραπάνω πρόγραμμα αλλά αυτή τη φορά με την χρήση ενός map. Θα έχει σαν key την λέξη και σαν value τον αριθμό εμφανίσεων αυτής της λέξης.

## Άσκηση 7

Να δημιουργήσετε ένα πρόγραμμα που θα έχει ένα map με key int και value vector από strings. Το πρόγραμμα θα διαβάζει ένα αρχείο κειμένου και θα τοποθετεί την κάθε λέξη στο αντίστοιχο vector από strings όπου θα έχει σαν key το μέγεθος της (π.χ. την λέξη "data" θα την τοποθετεί στο words[4]).