

ΠΡΟΟΔΟΣ Α'

A.M.:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

Θέμα 1

Να γράψετε κώδικα που να δημιουργεί μια δομή με όνομα `movie` (ταινία) και πεδία `title` (τίτλος-string), `year` (έτος-int) και `score` (βαθμολογία-double). Δημιουργήστε ένα διάνυσμα (vector της STL) που να περιέχει τις ακόλουθες 4 εγγραφές:

Τίτλος: 12 Angry Men, Έτος: 1957, Βαθμός: 8.9

Τίτλος: Inception, Έτος: 2010, Βαθμός: 8.7

Τίτλος: Blade Runner, Έτος: 1982, Βαθμός: 8.2

Τίτλος: Fargo, Έτος: 1996, Βαθμός: 8.1

Βρείτε και εμφανίστε το μέσο όρο των βαθμολογιών. Ταξινομήστε τις εγγραφές σε αύξουσα σειρά βαθμολογίας και εμφανίστε όλες τις εγγραφές.

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

using namespace std;

struct movie {
    string title;
    int year;
    double score;
    bool operator<(const movie &other) const { return score < other.score; }
};

int main(void) {
    vector<movie> movies = {"12 Angry Men", 1957, 8.9},
                        {"Inception", 2010, 8.7},
                        {"Blade Runner", 1982, 8.2},
                        {"Fargo", 1996, 8.1}};

    double sum = 0.0;
    for (movie m : movies)
        sum += m.score;
    cout << "Average score:" << sum / movies.size() << endl;

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

    for (movie m : movies)
        cout << m.title << " - " << m.year << " - " << m.score << endl;
}
```

Θέμα 2

Γράψτε συνάρτηση με όνομα fun που να δέχεται ως παραμέτρους δύο ταξινομημένους σε φθίνουσα σειρά πίνακες ακεραίων n θέσεων και να επιστρέφει έναν νέο πίνακα με ταξινομημένες τις n μεγαλύτερες τιμές και από τους δύο πίνακες. Καλέστε τη συνάρτηση στη main με ορίσματα δύο ταξινομημένους σε φθίνουσα σειρά, πίνακες 5 θέσεων .

Για παράδειγμα η κλήση της συνάρτησης για τους πίνακες {20, 15, 11, 8, 7} και {22, 12, 10, 2, 1} θα πρέπει να επιστρέψει {22, 20, 15, 12, 11}.

```
#include <iostream>

using namespace std;

void fun(int a[], int b[], int n, int c[]) {
    int k1 = 0, k2 = 0;
    for (int i = 0; i < n; i++)
        if (a[k1] > b[k2])
            c[i] = a[k1++];
        else
            c[i] = b[k2++];
}

int main(void) {
    int a[5] = {20, 15, 11, 8, 7};
    int b[5] = {22, 12, 10, 2, 1};
    int c[5];
    fun(a, b, 5, c);
    for (int i = 0; i < 5; i++)
        cout << c[i] << " ";
    cout << endl;
}
```

