

A.M.:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

Θέμα 1

Να γράψετε κώδικα που να δημιουργεί μια δομή με όνομα `customer` (πελάτης τράπεζας) που να περιέχει τα πεδία `name` (όνομα) και `balance` (υπόλοιπο λογαριασμού). Εισάγετε 3 πελάτες σε ένα διάνυσμα. Ταξινομήστε το διάνυσμα σε φθίνουσα σειρά υπολοίπου λογαριασμού και εμφανίστε τα περιεχόμενά του. Με τη χρήση `lambda` συνάρτησης και της συνάρτησης `count_if` εμφανίστε πόσοι από τους πελάτες έχουν υπόλοιπο λογαριασμού μηδέν.

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

using namespace std;

struct customer {
    string name;
    double balance;
    bool operator<(const customer &other) { return balance > other.balance; }
};

int main(void) {
    vector<customer> customers = {
        {"nikos", 2000.0}, {"maria", 500.0}, {"christos", 0.0}};
    sort(customers.begin(), customers.end());
    for (customer cust : customers)
        cout << cust.name << " " << cust.balance << endl;

    cout << "Customers having zero balance: "
         << count_if(customers.begin(), customers.end(),
                    [](customer c) { return c.balance == 0.0; })
         << endl;
}
```

Θέμα 2

Δημιουργήστε ένα map (λεξικό) με όνομα products που να περιέχει για υποθετικά προϊόντα ενός super market ζεύγη key-value της μορφής: κωδικός προϊόντος (string) – τιμή προϊόντος (double). Εισάγετε 5 προϊόντα της επιλογής σας στο λεξικό. Γράψτε κώδικα που ο χρήστης να εισάγει κωδικούς προϊόντων μέχρι να εισάγει το κενό. Για κάθε κωδικό προϊόντος που εισάγεται θα πρέπει να ελέγχεται αν υπάρχει στο λεξικό και εφόσον αυτό ισχύει να το προσθέτει σε ένα vector αλλιώς να εμφανίζει ένα μήνυμα λάθους και να συνεχίζει. Τέλος, όταν το πρόγραμμα τερματίζει θα πρέπει να εμφανίζει τη λίστα των κωδικών προϊόντων που εισήχθησαν και το συνολικό κόστος τους.

```
#include <iostream>
#include <map>
#include <vector>
#include <string>

using namespace std;

int main(void) {
    map<string, double> products = {
        {"A", 2.5}, {"B", 3.0}, {"C", 0.8}, {"D", 0.5}, {"E", 1.5}};

    double total = 0.0;
    vector<string> v;
    string code;
    cout << "Enter product code:";
    getline(cin, code); // in order to capture the empty string
    while (code != "") {
        if (products.find(code) == products.end())
            cout << "No such product code" << endl;
        else {
            cout << "OK" << endl;
            total += products[code];
            v.push_back(code);
        }
        cout << "Enter product code:";
        getline(cin, code);
    }
    for (string s : v)
        cout << s << " ";
    cout << "\nTotal:" << total << endl;
}
```