ΠΡΟΟΔΟΣ Γ’

Α.Μ.:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

Θέμα 1 [4 μονάδες]

1. Δηλώστε μια δομή customer (πελάτης) με πεδία name (όνομα, std::string) και balance (υπόλοιπο λογαριασμού, double).
2. Γράψτε μια συνάρτηση με όνομα hash_customer που να δέχεται ως παράμετρο μια εγγραφή πελάτη και να επιστρέφει το άθροισμα από τις τιμές που υπολογίζει η std::hash της STL για τα δύο πεδία της εγγραφής.
3. Γράψτε πρόγραμμα που ο χρήστης να εισάγει τα στοιχεία 10 πελατών και να αποθηκεύει τον κάθε πελάτη στη θέση που προσδιορίζει η συνάρτηση κατακερματισμού σε έναν πίνακα 10.001 θέσεων που δέχεται στοιχεία τύπου customer.

```
#include <iostream>

using namespace std;

struct customer {
        string name;
        double balance;
};

size_t hash_customer(struct customer &cust){
        hash<double> d_hash;
        hash<string> s_hash;
        return s_hash(cust.name) + d_hash(cust.balance);
}

int main(){
        constexpr int N = 10;
        struct customer hash_table[10001];
        for(int i=0;i<N;i++){
                struct customer acustomer;
                cout << "Enter customer's name and balance: ";
                cin  >> acustomer.name >> acustomer.balance;
                size_t h = hash_customer(acustomer) % 10001;
                cout << "customer inserted at index " << h << endl;
                hash_table[h] = acustomer;
        }
}
```

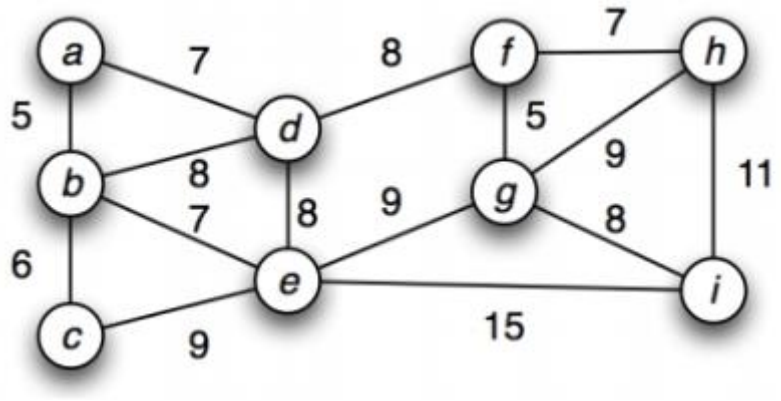
Θέμα 2 [3 μονάδες]

1. Στο γράφημα του θέματος 3 λάβετε υπόψη μόνο το υπογράφημα που σχηματίζεται από τις κορυφές a,b,c,d και e και γράψτε κώδικα που να δημιουργεί το υπογράφημα χρησιμοποιώντας ως αναπαράσταση τη λίστα γειτνίασης του εργαστηρίου 8 και χωρίς να γίνεται η ανάγνωση από αρχείο.
2. Συμπληρώστε τον κώδικα έτσι ώστε για κάθε κορυφή να εμφανίζει το μέσο όρο των βαρών των ακμών για τις κορυφές με τις οποίες συνδέεται απευθείας.
3. Συμπληρώστε τον κώδικα έτσι ώστε για κάθε περίπτωση κορυφής που δεν συνδέεται απευθείας με κάποια άλλη κορυφή να εισάγεται επιπλέον ακμή που να τις συνδέει με βάρος 1.

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

using namespace std;

int main() {
  // question 1
  map<string, vector<pair<int, string>>> graph;
  vector<pair<int, string>> va = {{5, "b"}, {7, "d"}};
  graph["a"] = va;
  vector<pair<int, string>> vb = {{5, "a"}, {8, "d"}, {6, "c"}, {7, "e"}};
  graph["b"] = vb;
  vector<pair<int, string>> vc = {{6, "b"}, {9, "e"}};
  graph["c"] = vc;
  vector<pair<int, string>> vd = {{7, "a"}, {8, "b"}, {8, "e"}, {8, "f"}};
  graph["d"] = vd;
  vector<pair<int, string>> ve = {
     {7, "b"}, {9, "c"}, {8, "d"}, {9, "g"}, {15, "i"}};
  graph["e"] = ve;

  // question 2
  vector<string> vertices = {"a", "b", "c", "d", "e"};
  for (auto &key : vertices) {
    double sum = 0.0;
    for (pair<int, string> p : graph[key]) {
      sum += p.first;
    }
    cout << "Vertex " << key << " average weight " << sum / graph[key].size()
         << endl;
  }

  // question 3
  for (auto &key1 : vertices) {
    for (auto &key2 : vertices) {
      if (key1 == key2)
        continue;
    bool found = false;
      for (pair<int, string> p : graph[key1]) {
        if (p.second == key2)
          found = true;
      }
      if (!found) {
        graph[key1].push_back(make_pair(1, key2));
        graph[key2].push_back(make_pair(1, key1));
        cout << "Edge added between vertex " << key1 << " and vertex " << key2
             << endl;
      }
    }
  }
}
```

Θέμα 3 [3 μονάδες]

1. Εφαρμόστε στο ακόλουθο γράφημα τον αλγόριθμο του Dijkstra για την εύρεση των συντομότερων διαδρομών προς όλες τις κορυφές χρησιμοποιώντας ως αφετηρία την κορυφή **i**.
2. Καταγράψτε τη συντομότερη διαδρομή για κάθε κορυφή και το μήκος της.



S	a	b	c	d	e	f	g	h	i
{}	INF	INF	INF	INF	INF	INF	INF	INF	0
{i}	INF	INF	INF	INF	15_i	INF	8_i	11_i	0
{i,g}	INF	INF	INF	INF	15_i	13_g	8_i	11_i	0
{i,g,h}	INF	INF	INF	INF	15_i	13_g	8_i	11_i	0
{i,g,h,f}	INF	INF	INF	21_f	15_i	13_g	8_i	11_i	0
{i,g,h,f,e}	INF	22_e	24_e	21_f	15_i	13_g	8_i	11_i	0
{i,g,h,f,e,d}	28_d	22_e	24_e	21_f	15_i	13_g	8_i	11_i	0
{i,g,h,f,e,d,b}	27_b	22_e	24_e	21_f	15_i	13_g	8_i	11_i	0
{i,g,h,f,e,d,b,c}	27_b	22_e	24_e	21_f	15_i	13_g	8_i	11_i	0
{i,g,h,f,e,d,b,c,a}	27_b	22_e	24_e	21_f	15_i	13_g	8_i	11_i	0

Αφετηρία: i
Shortest path from vertex i to vertex a is {i e b a} having length 27
Shortest path from vertex i to vertex b is {i e b} having length 22
Shortest path from vertex i to vertex c is {i e c} having length 24
Shortest path from vertex i to vertex d is {i g f d} having length 21
Shortest path from vertex i to vertex e is {i e} having length 15
Shortest path from vertex i to vertex f is {i g f} having length 13
Shortest path from vertex i to vertex g is {i g} having length 8
Shortest path from vertex i to vertex h is {i h} having length 11
Shortest path from vertex i to vertex i is {i} having length 0