

### Θέμα 1

Γράψτε συνάρτηση που να δέχεται ως παράμετρο έναν δισδιάστατο πίνακα ακεραίων με 3 στήλες και έναν αριθμό γραμμής και να επιστρέφει το άθροισμα των στοιχείων της γραμμής και το μικρότερο στοιχείο της γραμμής. Καλέστε τη συνάρτηση από κύριο πρόγραμμα με παραμέτρους τον πίνακα με τιμές  $\{\{1, 10, 7\}, \{3, 2, 5\}, \{8, 6, 7\}, \{0, 2, 8\}\}$  και τη 2<sup>η</sup> γραμμή του (δηλαδή την  $\{3, 2, 5\}$ ). Εμφανίστε τις τιμές που επιστρέφει η συνάρτηση.

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d01.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d01.cpp)

### Θέμα 2

Ορίστε μια δομή με όνομα pixel (εικονοστοιχείο) που να αποτελείται από 3 ακέραιες τιμές (unsigned short) red, green και blue. Κατασκευάστε μια συνάρτηση με όνομα luminance (φωτεινότητα) που να δέχεται μια εγγραφή pixel και να επιστρέφει την τιμή της παράστασης  $0.2126 \cdot \text{red} + 0.7152 \cdot \text{green} + 0.0722 \cdot \text{blue}$ . Δημιουργήστε ένα vector και εισάγετε σε αυτό 4 pixels της επιλογής σας. Εμφανίστε τα pixels ταξινομημένα κατά luminance σε αύξουσα σειρά.

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d02.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d02.cpp)

### Θέμα 3

Γράψτε κύριο πρόγραμμα που να αρχικοποιεί έναν πίνακα ακεραίων 100 θέσεων με τους αριθμούς 1 έως και 100 σε φθίνουσα σειρά. Να ταξινομεί τον πίνακα σε αύξουσα σειρά. Να δέχεται μια τιμή από τον χρήστη. Σε περίπτωση που η τιμή είναι από 1 έως και 100 να τη διαγράφει από τον πίνακα και να ολισθαίνει όλα τα στοιχεία που είναι δεξιά της κατά 1 θέση προς τα αριστερά, ενώ στη θέση της τιμής 100 να τοποθετεί την τιμή -1. Να εμφανίζει τον πίνακα στη νέα του μορφή.

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d03.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d03.cpp)

### Θέμα 4

Γράψτε πρόγραμμα που να εισάγει σε ένα πίνακα κατακερματισμού ως κλειδιά τα ονόματα 3 παικτών (kostas, maria, nikos) και ως τιμές μηδενικά. Το πρόγραμμα να δέχεται 3 ζεύγη τιμών (όνομα παίκτη και πόντοι) και να ενημερώνει τον πίνακα κατακερματισμού έτσι ώστε κάθε νέα τιμή πόντων που διαβάζεται να προστίθεται στο σκορ του κατάλληλου παίκτη. Να εμφανίζει το σκορ του κάθε παίκτη.

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d04.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d04.cpp)

### Θέμα 5

Ορίστε τη δομή ενός κόμβου μια απλής συνδεδεμένης λίστας που να περιέχει ως δεδομένα ονόματα και τηλέφωνα επαφών (π.χ. "nikos", "6977123456"). Στο κύριο πρόγραμμα εισάγετε τα περιεχόμενα δύο πινάκων 5 θέσεων με ονόματα και τηλέφωνα στη λίστα χρησιμοποιώντας δυναμική δέσμευση μνήμης. Εμφανίστε διανύοντας τη λίστα τα περιεχόμενα της. Διαγράψτε τη λίστα από τη μνήμη. Υλοποιήστε εκ νέου τις ίδιες ενέργειες χρησιμοποιώντας τη δομή συνδεδεμένης λίστας της STL, `std::list`.

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d05.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d05.cpp)

## Θέμα 6

Δίνεται η ακόλουθη κωδικοποίηση με πίνακα γειτνίασης ενός μη κατευθυνόμενου γραφήματος με βάρη. Να γράψετε συνάρτηση που να δέχεται ένα όνομα κορυφής και να εμφανίζει όλες τις άλλες κορυφές που είναι απευθείας συνδεδεμένες με αυτή. Από το κύριο πρόγραμμα καλέστε τη συνάρτηση για όλες τις κορυφές του γραφήματος και εμφανίστε τα αποτελέσματα.

```
#define V 6

string vertices[V] = {"A", "B", "C", "D", "E", "F"};

int adj_matrix[V][V] = {
    {0, 2, 6, 0, 0, 0},
    {2, 0, 3, 1, 0, 0},
    {6, 3, 0, 4, 3, 0},
    {0, 1, 4, 0, 2, 10},
    {0, 0, 3, 2, 0, 5},
    {0, 0, 0, 10, 5, 0}};
```

Λύση: [https://github.com/chgogos/ceteiep\\_dsa/blob/master/prep/prep\\_d06.cpp](https://github.com/chgogos/ceteiep_dsa/blob/master/prep/prep_d06.cpp)