

**ΣΕΤ ΑΣΚΗΣΕΩΝ 2****ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012**

**Προθεσμία: 27/11/11, 22:00**

**Περιεχόμενα**

- [Διαβάστε πριν ξεκινήσετε](#)
- [Εκφώνηση άσκησης](#)
- [Οδηγίες αποστολής άσκησης](#)

**Πριν ξεκινήσετε (ΔΙΑΒΑΣΤΕ ΑΥΤΗ ΤΗΝ ΕΝΟΤΗΤΑ!!)**

Διαβάστε *ολόκληρη* την εκφώνηση προσεκτικά και “σχεδιάστε” το πρόγραμμά σας στο χαρτί.

Αποθηκεύστε το πρόγραμμά σας σε ένα αρχείο με όνομα hw2 .c

Μην προσπαθήσετε να γράψετε το πρόγραμμα μια κι έξω. Να υλοποιείτε μία-μία τις συναρτήσεις με τη μέθοδο που χρησιμοποιήσαμε στο φροντιστήριο (αρχικά όλες οι συναρτήσεις είναι άδειες). Μετά την υλοποίηση κάθε μίας συνάρτησης, την τεστάρετε εκτεταμένα, διορθώνετε ότι λάθη έχει και μετά προχωράτε στην επόμενη συνάρτηση. Μην παραλείπετε να αποθηκεύετε αντίγραφα του προγράμματός σας κάθε φορά που υλοποιείτε ένα μεγάλο κομμάτι, ώστε αν σε κάποιο επόμενο στάδιο καταστραφεί το αρχείο σας, θα έχετε μια παλιότερη έκδοση από την οποία μπορείτε να συνεχίσετε. Τις επιμέρους “εκδόσεις” να τις ονομάζετε με διαφορετικά ονόματα.

Η άσκηση βαθμολογείται ως προς την ορθότητα, καλή χρήση συναρτήσεων, καλή χρήση δομών ελέγχου κι επανάληψης, σωστή χρήση σταθερών, αναγνωσιμότητα και σχολιασμό. Επίσης, πρέπει να ακολουθήσετε ακριβώς τις προδιαγραφές κάθε συνάρτησης και η έξοδος του προγράμματός σας να είναι ακριβώς ίδια με την αναμενόμενη.

Μη διστάζετε να ζητήσετε βοήθεια! Μπορείτε να χρησιμοποιήσετε το forum προγραμματισμού (<http://inf-server.inf.uth.gr/courses/coding/>) και φυσικά email.

Η εργασία αυτή μπορεί να γίνει σε ομάδες μέχρι 2 ατόμων. Μπορείτε να συζητάτε τις ασκήσεις με συμφοιτητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιοδήποτε τρόπο.

**Ξεκινήστε νωρίς!** Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ό,τι περιμένετε.

Εκπρόθεσμες ασκήσεις δε γίνονται δεκτές.

## Άσκηση: Σταθμός διοδίων

### Εισαγωγή

Θα γράψετε ένα πρόγραμμα το οποίο χρησιμοποιείται από ένα πλήρως αυτοματοποιημένο σταθμό διοδίων για να συλλέξει στοιχεία για τα οχήματα που περνούν από το σταθμό. Το πρόγραμμα ανιχνεύει κι αποθηκεύει τον τύπο κάθε οχήματος που περνά, την πινακίδα και το ποσό που χρωστάει ο οδηγός (κάποιοι οδηγοί περνούν τα διόδια χωρίς να πληρώνουν όλο το ποσό που οφείλουν). Αφού έχει συλλέξει όλες τις πληροφορίες, εντοπίζει οχήματα που πέρασαν πολλαπλές φορές ώστε να κρατά μόνο μία καταχώρηση ανά όχημα και τέλος υπολογίζει κι εκτυπώνει στατιστικά στοιχεία για τις οφειλές των οχημάτων.

Ακολουθούν λεπτομερείς οδηγίες για το πώς πρέπει να λειτουργεί το πρόγραμμά σας και τι συναρτήσεις πρέπει να περιέχει.

### Δομές Δεδομένων

Το πρόγραμμά σας πρέπει να χρησιμοποιεί δύο πίνακες:

Ο πρώτος είναι διδιάστατος πίνακας χαρακτήρων σε κάθε γραμμή του οποίου αποθηκεύονται τα στοιχεία μιας πινακίδας αυτοκινήτου. Αυτά είναι : δύο ή τρία γράμματα, ένα κενό, τέσσερα ψηφία, ένα κενό, κι ένας χαρακτήρας (C, B, T, ή M) ο οποίος αναπαριστά τον τύπο του αυτοκινήτου.

Ο δεύτερος είναι ένας μονοδιάστατος πίνακας αριθμών κινητής υποδιαστολής ο οποίος σε κάθε κελί  $i$  περιέχει το ποσό που χρωστάει ο οδηγός του οχήματος που βρίσκεται στη θέση  $i$  του διδιάστατου πίνακα. Αν ο οδηγός είχε πληρώσει όλο το αντίτιμο ή περισσότερο τότε αποθηκεύεται η τιμή 0, διαφορετικά αποθηκεύεται το ποσό που χρωστάει.

Επειδή δε γνωρίζουμε εκ των προτέρων πόσα αυτοκίνητα θα περάσουν από τα διόδια, οι πίνακες πρέπει να έχουν αρκετά μεγάλο μέγεθος (τουλάχιστον 10000). Πρέπει να σκεφτείτε τι τιμή είναι κατάλληλη για να αποθηκευτεί σε αχρησιμοποίητες θέσεις του πίνακα ώστε να σηματοδοτεί ότι είναι "άδειες".

Συμβουλή: αρχικά δώστε μικρό μέγεθος στους πίνακες για να μπορείτε να ελέγξετε εύκολα το πρόγραμμά σας, αλλά πριν παραδώσετε την άσκηση αλλάξτε το.

### Στάδια προγράμματος

Το πρόγραμμά σας πρέπει να αποτελείται από τα παρακάτω στάδια, τα οποία θα αναλυθούν σε επόμενες ενότητες:

1. Διαβάζει τα δεδομένα από το πληκτρολόγιο και τα αποθηκεύει στους πίνακες που περιγράφονται στην προηγούμενη ενότητα.
2. Ταξινομεί τα περιεχόμενα των πινάκων ως προς τις πινακίδες (λεξικογραφικά), σε αύξουσα σειρά
3. Απαλείφει πολλαπλές καταχωρήσεις ανά όχημα και ανανεώνει αναλόγως τις οφειλές
4. Εκτυπώνει τα στοιχεία των οχημάτων και τις αντίστοιχες οφειλές
5. Υπολογίζει κι εκτυπώνει στατιστικά στοιχεία

### 1: Ανάγνωση και αποθήκευση δεδομένων

Για την ανάγνωση και αποθήκευση δεδομένων, γράψτε μια συνάρτηση η οποία παίρνει ως παραμέτρους ένα διδιάστατο πίνακα χαρακτήρων κι ένα μονοδιάστατο πίνακα αριθμών κινητής υποδιαστολής και δεν επιστρέφει τίποτα.

Η συνάρτηση διαβάζει από το πληκτρολόγιο ένα χαρακτήρα ο οποίος εκφράζει τον τύπο ενός οχήματος. Ο τύπος μπορεί να είναι ένας από τους παρακάτω:

- M ή m για μοτοσικλέτα
- C ή c για αυτοκίνητο
- B ή b για λεωφορείο
- T ή t για φορτηγό

Ακολούθως, διαβάζει τα γράμματα μιας πινακίδας, τα οποία είναι δύο ή τρία και μετά τα ψηφία τα οποία είναι πάντα τέσσερα. Προσέξτε: τα ψηφία πρέπει να διαβαστούν ως συμβολοσειρά και όχι ως ακέραιος (γιατί μπορεί να ξεκινούν από το μηδέν). Τέλος, διαβάζει ένα αριθμό κινητής υποδιαστολής ο οποίος αναπαριστά το ποσό που πλήρωσε το συγκεκριμένο αυτοκίνητο.

Μπορείτε να υποθέσετε ότι η είσοδος θα είναι πάντα έγκυρη.

Στον πρώτο πίνακα αποθηκεύεται η πινακίδα ως εξής: τα γράμματα, ένα κενό, οι αριθμοί, ένα κενό και τέλος ένας από τους χαρακτήρες 'M', 'C', 'B', ή 'T' ανάλογα με τον τύπο του οχήματος. Σημειώστε πως ο τύπος πρέπει να εμφανίζεται με κεφαλαίο γράμμα στην πινακίδα, ακόμη κι αν στην είσοδο δόθηκε ως μικρό. Προσέξτε να τερματίζει σωστά σε '\0' η συμβολοσειρά σας.

Στην αντίστοιχη θέση του δεύτερου πίνακα πρέπει να αποθηκευτεί το ποσό που οφείλει ο οδηγός. Έχετε διαβάσει πόσο πλήρωσε, και γνωρίζετε από το hw1 πόσο είναι το αντίτιμο για κάθε τύπο οχήματος. Σας θυμίζουμε:

- €1.25 για μοτοσικλές
- €2.35 για αυτοκίνητα
- €8.40 για φορτηγά
- €16.50 για λεωφορεία

Αν ο οδηγός πλήρωσε ακριβώς το ποσό που πρέπει, ή πλήρωσε παραπάνω (οπότε εννοείται ότι πήρε ρέστα αν και το πρόγραμμά μας δεν ασχολείται με αυτό), τότε χρωστάει 0 ευρώ. Αν πλήρωσε λιγότερα, τότε χρωστάει τη διαφορά.

## **2: Ταξινόμηση**

Γράψτε μια συνάρτηση η οποία χρησιμοποιεί τον αλγόριθμο selection sort για να ταξινομήσει τα στοιχεία των οχημάτων σε αύξουσα σειρά, λεξικογραφικά με βάση τις πινακίδες. Μην ξεχάσετε ότι ο πίνακας των οφειλών πρέπει πάντα να διατηρείται έτσι ώστε στη θέση i να είναι το ποσό που οφείλει το όχημα που βρίσκεται στη θέση i του πίνακα πινακίδων.

## **3: Απαλοιφή πολλαπλών καταχωρήσεων**

Τώρα που οι πινακίδες είναι ταξινομημένες, αν υπάρχουν πολλαπλές καταχωρήσεις για το ίδιο όχημα, θα πρέπει να είναι σε διαδοχικές θέσεις του πίνακα. Σκοπός μας είναι να σβήσουμε όλες τις διπλο-καταχωρήσεις εκτός από μία, στην οποία θα αντιστοιχεί το *συνολικό* ποσό που οφείλει αυτό το αυτοκίνητο.

Ας υποθέσουμε πως έχετε εντοπίσει ότι οι καταχωρήσεις ανάμεσα στις θέσεις start και end αναφέρονται όλες στο ίδιο όχημα. Γράψτε αρχικά μια βοηθητική συνάρτηση η οποία παίρνει ως παραμέτρους τον πίνακα οφειλών και τις θέσεις start και end κι επιστρέφει το άθροισμα των οφειλών για αυτές τις θέσεις.

Ακολούθως, γράψτε μια συνάρτηση που διατρέχει τον πίνακα πινακίδων, εντοπίζει διαδοχικές θέσεις με την ίδια πινακίδα και τις σβήνει όλες εκτός από μία. Το ίδιο φυσικά γίνεται και στον πίνακα οφειλών στον οποίο όμως θα αποθηκευτεί το σύνολο των οφειλών (χρησιμοποιήστε τη βοηθητική συνάρτηση).

Μπορείτε είτε να μαρκάρετε τις κενές θέσεις ως "άδειες" ή να μετακινήσετε τα επόμενα κελιά στα αριστερά, όσες θέσεις χρειάζεται.

#### **4: Εκτύπωση στοιχείων**

Γράψτε μια συνάρτηση η οποία εκτυπώνει τα περιεχόμενα των πινάκων ως εξής:

Για κάθε όχημα εκτυπώνει:

Την πινακίδα όπως αυτή έχει αποθηκευτεί στον πίνακα πινακίδων, και έτσι ώστε να καταλαμβάνει 11 θέσεις, ένα κόμμα, ένα κενό, το αντίστοιχο οφειλόμενο ποσό με δύο δεκαδικά ψηφία, ένα χαρακτήρα αλλαγής γραμμής.

Στο τέλος εκτυπώνει σε νέα γραμμή μια σειρά από 20 χαρακτήρες = και ένα χαρακτήρα αλλαγής γραμμής.

Δεν πρέπει να εκτυπώνεται τίποτα για "άδειες" θέσεις του πίνακα.

#### **5: Υπολογισμός στατιστικών**

Γράψτε μια βοηθητική συνάρτηση η οποία υπολογίζει κι επιστρέφει το συνολικό ποσό που οφείλουν όλα τα οχήματα.

Γράψτε μια βοηθητική συνάρτηση η οποία παίρνει ως παράμετρο τον τύπο ενός οχήματος και ό,τι άλλο χρειάζεται και υπολογίζει κι επιστρέφει το μέσο οφειλόμενο ποσό ανά όχημα τέτοιου τύπου επί του συνόλου των οχημάτων αυτού του τύπου που χρωστούν ποσό μεγαλύτερο του μηδενός.

Γράψτε μια συνάρτηση η οποία χρησιμοποιεί τις παραπάνω βοηθητικές συναρτήσεις για να εκτυπώσει τα παρακάτω μηνύματα:

Total amount owed: V euros

Motorcycles owe X euros on average.

Cars owe W euros on average.

Buses owe Y euros on average.

Trucks owe Z euros on average.

όπου V το συνολικό οφειλόμενο ποσό, X το μέσο ποσό για μοτοσυκλέτες, W το μέσο ποσό για αυτοκίνητα, Y το μέσο ποσό για λεωφορεία και Z το μέσο ποσό για φορτηγά. Όλα τα ποσά πρέπει να έχουν δύο δεκαδικά ψηφία.

#### **Έλεγχος ορθότητας**

Το πρόγραμμά σας πρέπει να λειτουργεί σωστά και να εκτυπώνει όλα τα μηνύματα και αποτελέσματα ακριβώς με τον τρόπο που σας περιγράφουμε. Για να μπορέσετε να ελέγξετε την ορθότητα θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου. Υπάρχει ένας εύκολος τρόπος να συγκρίνετε τα δικά σας αποτελέσματα με τα δικά μας:

Ας υποθέσουμε ότι το εκτελέσιμο πρόγραμμά σας λέγεται hw2, το ενδεικτικό αρχείο εισόδου λέγεται `input1.txt` και το αντίστοιχο αρχείο εξόδου που σας έχουμε δώσει λέγεται `stdout1.txt`

Η εντολή:

```
./hw1 < input1.txt > myout1.txt
```

εκτελεί το πρόγραμμά σας με την ενδεικτική είσοδο `input1.txt` και αποθηκεύει τα αποτελέσματα στο αρχείο εξόδου `myout1.txt`

Η εντολή

```
diff -b myout1.txt stdout1.txt
```

συγκρίνει το δικό σας αρχείο εξόδου με το δικό μας. Αν υπάρχουν διαφορές, τις εμφανίζει (γραμμή-γραμμή). Αν δεν υπάρχουν διαφορές, δεν κάνει τίποτα.

Για περισσότερες πληροφορίες δείτε τα παραρτήματα A και B του φυλλαδίου για το hw1.

Πρέπει το πρόγραμμα που θα μας παραδώσετε να παράγει έξοδο που δεν έχει διαφορές από τη δική μας.

**Πριν παραδώσετε το πρόγραμμά σας, προσθέστε σε σχόλια στην αρχή του αρχείου τα πλήρη ονόματα και ΑΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με λατινικούς χαρακτήρες.**

Απαγορεύεται αυστηρά η χρήση καθολικών μεταβλητών.

**Σημείωση:** Αν ξεκινήσετε το πρόγραμμα πριν τη διάλεξη της Τετάρτης 16/11 όπου θα μάθετε για πίνακες ως παραμέτρους συναρτήσεων, μπορείτε *προσωρινά* να ορίσετε τους πίνακες ως καθολικές μεταβλητές. ΑΛΛΑ μετά θα πρέπει να διορθώσετε το πρόγραμμά σας ώστε να τους περνάτε ως παραμέτρους όπου χρειάζεται. Το πρόγραμμα που θα παραδώσετε τελικά δεν πρέπει να περιέχει ούτε μια καθολική μεταβλητή.

Αρχεία προς παράδοση: hw2.c

## Πώς να παραδώσετε τη δουλειά σας

(Ακολουθείστε τις οδηγίες ακριβώς αλλιώς μπορεί να μη δούμε τα αρχεία σας)

Κατασκευάστε ένα φάκελο με όνομα epwnumero1\_AM1\_epwnumero2\_AM2 και αντιγράψτε μέσα σε αυτόν το hw2.c

Πηγαίνετε στο φάκελο μέσα στον οποίο βρίσκεται το epwnumero1\_AM1\_epwnumero2\_AM2 που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czf epwnumero1_AM1_epwnumero2_AM2.tar.gz epwnumero1_AM1_epwnumero2_AM2
```

Στείλτε email:

- στη διεύθυνση **ce120lab@gmail.com**
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **CE120 hw2**
- και επικολλημένο αρχείο το `epwnumero1_AM1_epwnumero2_AM2.tar.gz`