ΣΕΤ ΑΣΚΗΣΕΩΝ 3

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012

Προθεσμία: 8/12/11, 22:00

Περιεχόμενα

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 `hw3.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, σωστή χρήση struct και, αν χρειάζεται, enum, 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

Στο δεύτερο σετ ασκήσεων χρειάστηκε να αποθηκεύετε για κάθε αυτοκίνητο τον τύπο, την πινακίδα και το ποσό που χρωστάει. Επειδή αυτές οι ποσότητες είναι διαφορετικού τύπου (χαρακτήρας, συμβολοσειρά και αριθμός κινητής υποδιαστολής), αναγκαστήκατε πρώτα να ενώσετε κάθε τύπο με την αντίστοιχη πινακίδα σε μία συμβολοσειρά και μετά να χρησιμοποιήσετε δύο "παράλληλους" πίνακες, πράγμα που κάνει τον κώδικα πιο μπερδεμένο και αυξάνει την πιθανότητα λάθους (πχ να γίνει κάποια αλλαγή στον ένα πίνακα και να μη γίνει αντίστοιχη αλλαγή στον άλλο).

Όταν θέλουμε να αποθηκεύσουμε δεδομένα διαφορετικών τύπων που όμως αναφέρονται στην ίδια οντότητα (πχ σε ένα όχημα), ο πιο σωστός τρόπος είναι να ομαδοποιήσουμε αυτούς τους τύπους σε ένα struct.

Για το τρίτο σετ ασκήσεων, θα επαναλάβετε το δεύτερο, αλλά αυτή τη φορά αντί για δύο πίνακες δεδομένων (έναν από συμβολοσειρές κι έναν από αριθμούς), θα έχετε έναν πίνακα από struct.

Επιπλέον, θα υλοποιήσετε δύο ακόμη αλγορίθμους ταξινόμησης, θα τρέξετε το πρόγραμμά σας για κάθε έναν και θα μαζέψετε στατιστικά στοιχεία σχετικά με την απόδοση των τριών αλγορίθμων.

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

Δομές Δεδομένων

Κατασκευάστε ένα struct το οποίο θα χρησιμοποιηθεί για την αναπαράσταση ενός οχήματος όπως αυτό είναι αποθηκευμένο στον πίνακα οχημάτων. Για το σκοπό αυτό, το struct πρέπει να περιέχει πεδία για:

- Τον τύπο του οχήματος (χαρακτήρας)
- Την πινακίδα του οχήματος (συμβολοσειρά με 2 ή 3 γράμματα, κενό, 4 ψηφία)
- Το ποσό που χρωστά το συγκεκριμένο όχημα (αριθμός κινητής υποδιαστολής)
- Αν το συγκεκριμένο όχημα είναι έγκυρο (boolean ποσότητα)

Το τελευταίο πεδίο χρειάζεται ώστε να γνωρίζουμε αν στο συγκεκριμένο κελί του πίνακα βρίσκεται έγκυρο αυτοκίνητο ή θεωρείται ότι το κελί είναι κενό.

Το πρόγραμμά σας πρέπει να χρησιμοποιεί ένα πίνακα από struct στον οποίο αποθηκεύονται όλα τα στοιχεία ενός οχήματος. Αυτός ο πίνακας ουσιαστικά αντικαθιστά τους δύο πίνακες (ποσών και πινακίδων) που χρησιμοποιήσατε στο σετ 2.

Επειδή δε γνωρίζουμε εκ των προτέρων πόσα αυτοκίνητα θα περάσουν από τα διόδια, ο πίνακας πρέπει να έχουν αρκετά μεγάλο μέγεθος (τουλάχιστον 10000).

Στάδια προγράμματος

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

1. Διαβάζει από τη γραμμή εντολής ένα ακέραιο αριθμό ο οποίος προσδιορίζει ποιος αλγόριθμος ταξινόμησης θα χρησιμοποιηθεί.
2. Διαβάζει τα δεδομένα από το πληκτρολόγιο και τα αποθηκεύει στον πίνακα που περιγράφεται στην προηγούμενη ενότητα.
3. Ταξινομεί τα περιεχόμενα του πίνακα ως προς τις πινακίδες (λεξικογραφικά), σε αύξουσα σειρά
4. Απαλείφει πολλαπλές καταχωρήσεις ανά όχημα και ανανεώνει αναλόγως τις οφειλές
5. Εκτυπώνει τα στοιχεία των οχημάτων και τις αντίστοιχες οφειλές
6. Υπολογίζει κι εκτυπώνει στατιστικά στοιχεία
7. Υπολογίζει και εκτυπώνει στοιχεία για το χρόνο εκτέλεσης της ταξινόμησης

1: Εισαγωγή κωδικού αλγορίθμου ταξινόμησης

Το πρόγραμμά σας πρέπει να ελέγχει αν ο χρήστης προσδιόρισε επιπλέον όρισμα στη γραμμή εντολής. Αν ναι, τότε ελέγχει αν αυτό ήταν ο αριθμός 1 ή 2 ή 3 και τον αποθηκεύει σε μια μεταβλητή. Αν ήταν οποιοσδήποτε άλλος αριθμός ή δεν είχε προσδιοριστεί όρισμα, τότε αποθηκεύεται στη μεταβλητή το 1.

2: Ανάγνωση και αποθήκευση δεδομένων

Επαναλάβετε το στάδιο 1 του σετ 2, έχοντας αντικαταστήσει τους δύο πίνακες του σετ 2 με το νέο πίνακα από struct. Αυτή τη φορά, ο τύπος του οχήματος αποθηκεύεται χρησιμοποιώντας το κατάλληλο πεδίο και όχι ως μέρος της πινακίδας. Κατά τα άλλα, η αρχικοποίηση είναι ακριβώς ίδια με το σετ 2. Θεωρείστε πως η ανάγνωση οχημάτων σταματά όταν δοθεί τύπος '0'.

3: Ταξινόμηση

Μεταβάλετε τη συνάρτηση ταξινόμησης με selection sort έτσι ώστε να χρησιμοποιεί το νέο πίνακα.

Γράψτε μια συνάρτηση η οποία χρησιμοποιεί τον αλγόριθμο insertion sort για να ταξινομήσει τα στοιχεία των οχημάτων σε αύξουσα σειρά, λεξικογραφικά με βάση τις πινακίδες.

Γράψτε μια συνάρτηση η οποία χρησιμοποιεί τον αλγόριθμο bubble sort για να ταξινομήσει τα στοιχεία των οχημάτων σε αύξουσα σειρά, λεξικογραφικά με βάση τις πινακίδες.

Στη main θα πρέπει να ελέγχετε την τιμή της μεταβλητής που περιγράφεται στο στάδιο 1, και αναλόγως να καλείτε την κατάλληλη συνάρτηση (1: insertion sort, 2: selection sort, 3:bubble sort).

4: Απαλοιφή πολλαπλών καταχωρήσεων

Επαναλάβετε το αντίστοιχο στάδιο του σετ 2, χρησιμοποιώντας το νέο πίνακα.

5: Εκτύπωση στοιχείων

Επαναλάβετε το αντίστοιχο στάδιο του σετ 2, χρησιμοποιώντας το νέο πίνακα.

6: Υπολογισμός στατιστικών

Επαναλάβετε το αντίστοιχο στάδιο του σετ 2, χρησιμοποιώντας το νέο πίνακα.

7: Υπολογισμός χρόνου εκτέλεσης

Χρησιμοποιείστε τη συνάρτηση gettimeofday στη main για να καταγράψετε την ώρα ακριβώς πριν και ακριβώς μετά την κλήση στη συνάρτηση ταξινόμησης. Υπολογίστε πόσα microseconds διήρκεσε η εκτέλεση του αλγορίθμου, κι εκτυπώστε το μήνυμα "TIME TO SORT: ?" όπου ? ο αριθμός microsecond που υπολογίσατε. Μετά το μήνυμα εκτυπώστε χαρακτήρα αλλαγής γραμμής.

Εκτελέστε το πρόγραμμά σας για αρχεία εισόδου με 10, 1000 και 10000 οχήματα, για κάθε έναν αλγόριθμο, και καταγράψτε τους χρόνους εκτέλεσης. Προσθέστε ένα μεγάλο σχόλιο στο τέλος του προγράμματός σας όπου θα καταγράψετε τα αποτελέσματά σας, τι συμπεράσματα βγάζετε από αυτά, και πού κατά τη γνώμη σας οφείλονται οι διαφορές στους χρόνους εκτέλεσης. Παρακαλούμε γράψτε με ΛΑΤΙΝΙΚΟΥΣ χαρακτήρες, όχι με ελληνικούς.

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 `hw3`, θέλουμε να χρησιμοποιήσουμε selection sort (επιλογή 2) το ενδεικτικό αρχείο εισόδου λέγεται `test1` και το αντίστοιχο αρχείο εξόδου που σας έχουμε δώσει λέγεται `test1.std`

Η εντολή:

```
./hw3 2 < test1 > test1.out
```

εκτελεί το πρόγραμμά σας με την ενδεικτική είσοδο `test1` και αποθηκεύει τα αποτελέσματα στο αρχείο εξόδου `test1.out`

Η εντολή `diff -b test1.out test1.std`

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 Α και Β του φυλλαδίου για το hw1.

Πρέπει το πρόγραμμα που θα μας παραδώσετε να παράγει έξοδο που δεν έχει διαφορές από τη δική μας, εκτός φυσικά από τους χρόνους εκτέλεσης.

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 <u>αυστηρά</u> η χρήση καθολικών μεταβλητών.

Αρχεία προς παράδοση: hw3.c

Πώς να παραδώσετε τη δουλειά σας

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

Κατασκευάστε ένα φάκελο με όνομα `epwnumo1_AM1_epwnumo2_AM2` και αντιγράψτε μέσα σε αυτόν το `hw3.c`

Πηγαίνετε στο φάκελο μέσα στον οποίο βρίσκεται το `epwnumo1_AM1_epwnumo2_AM2` που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czf epwnumo1_AM1_epwnumo2_AM2.tgz epwnumo1_AM1_epwnumo2_AM2
```

Στείλτε email:

- στη διεύθυνση **ce120lab@gmail.com**
- αντίγραφο (**CC**) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **CE120 hw3**
- και επικολλημένο αρχείο το epwnumo1_AM1_epwnumo2_AM2.tgz