

## Προγραμματιστική Άσκηση 4

Ημερομηνία Παράδοσης: Πέμπτη, 11 Δεκεμβρίου 2008

**Πρόβλημα 1 [10 μονάδες]** Δεδομένου ενός συνόλου,  $n$  το πλήθος, διατάξιμων αντικειμένων, δηλαδή  $n$  αντικειμένων που μπορώ να τα διατάξω/ταξινομήσω σε αύξουσα (ή και φθίνουσα) σειρά, ο **διάμεσος** είναι το μεσαίο αντικείμενο στην ταξινομημένη λίστα των αντικειμένων ή εναλλακτικά το αντικείμενο εκείνο που ισαπέχει στην ταξινομημένη λίστα από το ελάχιστο και το μέγιστο στοιχείο. Ο παραπάνω ορισμός είναι ακριβής στην περίπτωση που το πλήθος των αντικειμένων  $n$  είναι περιττός αριθμός, ενώ αντίθετα δεν είναι σωστός για  $n$  ζυγό. Στην περίπτωση αυτή υπάρχουν στην πραγματικότητα δύο διάμεσοι, εκ των οποίων εμείς κατά σύμβαση δια διαλέγουμε εκείνον που είναι κοντινότερα στο ελάχιστο στοιχείο.

Για παράδειγμα αν θεωρήσουμε το σύνολο των θετικών ακεραίων αριθμών  $\{5, 7, 10, 67, 34, 1\}$ , ο διάμεσος, σύμφωνα με τον παραπάνω ορισμό είναι ο αριθμός 7, ενώ για το σύνολο  $\{56, 33, 14, 21, 78\}$  ο διάμεσος είναι ο 33 (τα αντίστοιχα ταξινομημένα σύνολα είναι  $\{1, 5, 7, 10, 34, 67\}$  και  $\{14, 21, 33, 56, 78\}$ ).

Στη συγκεκριμένη προγραμματιστική άσκηση μας ενδιαφέρει ο υπολογισμός του διαμέσου ενός συνόλου από αλφαριθμητικά (από αντικείμενα τύπου `string`). Ο τρόπος με τον οποίο συγκρίνω `strings` είναι η συνάρτηση `StringCompare` της βιβλιοθήκης `strlib`, η οποία δηλώνεται ως:

```
int StringCompare(string s1, string s2);
```

και η οποία επιστρέφει αρνητική τιμή αν το αλφαριθμητικό `s1` είναι «μικρότερο» του `s2`, θετική τιμή αν το αλφαριθμητικό `s1` είναι «μεγαλύτερο» του `s2`, και μηδέν αν τα `s1` και `s2` είναι «ίσα» (δηλαδή το ίδιο αλφαριθμητικό γράμμα προς γράμμα). Η συνάρτηση `StringCompare` ουσιαστικά συγκρίνει τα αλφαριθμητικά χαρακτήρα προς χαρακτήρα από αριστερά προς τα δεξιά, χρησιμοποιώντας τους κώδικες ASCII των χαρακτήρων, και θεωρώντας ότι μικρότερα σε μήκος αλφαριθμητικά προηγούνται εκείνων με μεγαλύτερο μήκος. Για παράδειγμα αν έχω το σύνολο των αλφαριθμητικών  $\{\text{desk, pen, paper, ink, Letter, ruler, Pencil}\}$  η αντίστοιχη ταξινομημένη λίστα είναι η  $\{\text{Letter, Pencil, desk, ink, paper, pen, ruler}\}$ , ενώ ο διάμεσος είναι το αλφαριθμητικό `ink`. Αντίστοιχα για το σύνολο  $\{\text{Desk, DESK, DeSk, DESKS}\}$ , το ταξινομημένο σύνολο είναι το  $\{\text{DESK, DESKS, DeSk, Desk}\}$  και ο διάμεσος το αλφαριθμητικό `DESKS`.

Ακόμη πιο συγκεκριμένα, το ζητούμενο αυτής της προγραμματιστικής άσκησης είναι συμπληρώσετε τον κώδικα που σας δίνεται στο URL

<http://www.tem.uoc.gr/~mkaravel/Courses/EM101-fall08/hwks/hwk5-files.zip>

και ειδικότερα να συμπληρώσετε:

1. το κυρίως σώμα της συνάρτησης `StringMedian` στο αρχείο `StringMedian.c`, η οποία δεδομένου ενός πίνακα από `strings` του πραγματικού μεγέθους αυτού υπολογίζει τον διάμεσο και
2. τον κώδικα που υπολείπεται στη συνάρτηση `ReadWords` που βρίσκεται στο αρχείο `median.c` η οποία συνάρτηση διαβάζει λέξεις από αρχείο με δοσμένο όνομα και τις τοποθετεί σε έναν πίνακα που παίρνει ως τυπικό όρισμα. Η συνάρτηση `ReadWords` κατά την επιστροφή επιστρέφει τον αριθμό των λέξεων που διάβασε από το δοσμένο αρχείο.
3. τα στοιχεία σας υπό μορφή σχολίων στα δύο προαναφερθέντα αρχεία.

Γενικότερα, διαβάστε τα σχόλια των δύο αρχείων για να δείτε σε ποια σημεία αναμένεται να συμπληρώσετε κάτι. Πέραν των σημείων που σας υποδεικνύεται ότι πρέπει να συμπληρώσετε κάτι, προσθέστε ότι άλλο θεωρείτε εσείς αναγκαίο ή/και χρήσιμο.

Πέραν των αρχείων `StringMedian.c` και `median.c`, σας δίνονται επίσης τα εξής αρχεία:

StringMedian.h Header file στο οποίο δηλώνεται η συνάρτηση StringMedian.

FileReader.h Header file στο οποίο δηλώνονται συναρτήσεις που μας επιτρέπουν να διαβάζουμε λέξεις από αρχείο. Διαβάστε τα σχόλια στο αρχείο αυτό για να καταλάβετε ποιες συναρτήσεις σας παρέχονται και τί κάνουν.

FileReader.c Αρχείο C στο οποίο δίνονται οι υλοποιήσεις των συναρτήσεων που δηλώθηκαν στο αρχείο FileReader.h.

example1.dat Αρχείο με λέξεις που μπορείτε να χρησιμοποιήσετε για να ελέγξετε το πρόγραμμά σας.

example2.dat Ένα ακόμη αρχείο με λέξεις που μπορείτε να χρησιμοποιήσετε για να ελέγξετε το πρόγραμμά σας.

Το κυρίως πρόγραμμα που σας δίνεται ήδη υλοποιημένο (βλέπε συνάρτηση main) ζητάει από το χρήστη να δώσει όνομα αρχείου από το οποίο να διαβάσει λέξεις. Συνέχεια, η συνάρτηση ReadWords διαβάζει τις λέξεις από το αρχείο αυτό και μετά εκτυπώνονται στην οθόνη το πλήθος των λέξεων που διαβάστηκαν, αλλά και ο διάμεσος του συνόλου των λέξεων.

Παρακάτω σας δίνεται ενδεικτική συμπεριφορά του προγράμματος που πρέπει να γράψετε (τα δεδομένα που δίνει ο χρήστης γράφονται με κόκκινο, το prompt με μαύρο ενώ ότι εκτυπώνει το πρόγραμμα με μπλε). Στο παρακάτω παράδειγμα τρέχουμε το εκτελέσιμο αρχείο median δύο φορές, δίνοντας αντίστοιχα τα αρχεία example1.dat και example2.dat.

```
[my prompt]$ ./median ←
Enter file containing words: example1.dat ←
Number of words: 10
The median word is: "intramural"
[my prompt]$ ./median ←
Enter file containing words: example2.dat ←
Number of words: 33
The median word is: "documentation"
[my prompt]$
```

Στα αρχεία StringMedian.c και median.c, τα οποία είναι και τα μόνα που θα τροποποιήσετε, θα πρέπει να συμπληρώσετε στα ήδη υπάρχοντα σχόλια το ονοματεπώνυμό σας και τον αριθμό μητρώου σας, καθώς και ότι άλλες πληροφορίες είναι σχόπιμες προκειμένου όποιος διαβάσει το πρόγραμμά σας να καταλάβει τί γίνεται. Δεδομένου ότι στη συγκεκριμένη άσκηση θα δουλέψετε με παραπάνω από ένα αρχεία, στο zip αρχείο που σας δίνεται τα αρχεία είναι εντός του directory hwk5-files. Σας προτίνεται να δουλέψετε εντός του directory αυτού.

Για τη μεταγλώττιση του προγράμματός σας μπορείτε να ακολουθήσετε τις οδηγίες που σας έχουν δοθεί στις διαλέξεις και τα εργαστήρια, σχετικά με το πώς μεταγλωττίζουμε προγράμματα των οποίων ο κώδικας υπάρχει σε πολλά αρχεία. Εναλλακτικά, στο zip αρχείο hwk5-files.zip σας δίδεται ένα makefile αρχείο με όνομα Makefile με το οποίο μπορείτε να μεταγλωττίσετε το πρόγραμμά σας (όπως και στην Προγραμμασιστική Άσκηση 4).

Ο βαθμός σας για την άσκηση αυτή χωρίζεται στις εξής κατηγορίες:

- (α') [2 μονάδες] Την αναγνωσιμότητα του κώδικά σας: θα πρέπει να έχετε επαρκή σχόλια και να είναι καθαρογραμμένος ο κώδικάς σας.
- (β') [4 μονάδες] Την ορθότητα του κώδικά σας (θα πρέπει να δίνει τα σωστά αποτελέσματα).
- (γ') [2 μονάδες] Τήρηση των προδιαγραφών που αναφέρονται στην εκφώνηση.
- (δ') [2 μονάδες] Από την απλότητα και αποδοτικότητα του κώδικά σας.

Σύνολο μονάδων: 10