11.4 Άσκηση 4-Ταξινόμηση λέξεων

11.4.1 Περιγραφή

Να γραφεί πρόγραμμα το οποίο να παρέχει στο χρήστη διαμέσου μενού επιλογής τις παρακάτω δυνατότητες.

1. Εισαγωγή ενός αριθμού λέξεων μέγιστου πλήθους 30.
2. Εμφάνιση της λίστας των λέξεων.
3. Ταξινόμηση της λίστας των λέξεων κατά αύξουσα σειρά.
4. Ταξινόμηση της λίστας των λέξεων κατά φθίνουσα σειρά.
5. Εμφάνιση στατιστικών στοιχείων όπως:
   - μέγιστο μήκος λέξης,
   - ελάχιστο μήκος λέξης,
   - μέσο μήκος λέξης.
6. Εισαγωγή χαρακτήρα και ανεύρεση των παρακάτω στοιχείων:
   - συνολικό αριθμό εμφανίσεων του χαρακτήρα στις λέξεις της λίστας,
   - μέσο αριθμό εμφανίσεων ανά λέξη,
   - μέγιστο και ελάχιστο αριθμό επαναλήψεων για μία λέξη,
   - τη λέξη με το μεγαλύτερο αριθμό επαναλήψεων,
   - τη λέξη με το μικρότερο αριθμό επαναλήψεων.
7. Εισαγωγή λέξης και αναζήτηση της στην λίστα. Αν η λέξη υπάρχει, να εμφανίζεται κατάλληλο μήνυμα, αλλά και η σειρά της στον πίνακα. Αν δεν υπάρχει, να δίνεται η δυνατότητα εισαγωγής της στη λίστα. Η εισαγωγή θα επιτρέπεται μόνο εάν υπάρχει διαθέσιμος χώρος.

11.4.2 Στόχος

Εξοικείωση με:

- την έννοια της δόμησης προγράμματος με συναρτήσεις,
- την έννοια της τεχνικής της αυξητικής ανάπτυξης (incremental development),
- τη διαχείριση αλφαριθμητικών,
- τις συναρτήσεις ταξινόμησης και αναζήτησης.

Χρόνος εκτέλεσης: *Κατά τη διάρκεια της μελέτης του κεφαλαίου 7 ή με την ολοκλήρωσή του.*

11.4.3 Διαδικασία ανάπτυξης

Χρησιμοποιήστε την τεχνική της αυξητικής ανάπτυξης (**incremental development τεχνική**). Αναπτύξατε το πρόγραμμα σταδιακά ολοκληρώνοντας κάθε φορά ένα τμήμα του και ελέγχοντας την ορθότητα του τμήματος που αναπτύξατε. Αυτό γίνεται μεταγλωττίζοντας και εκτελώντας το πρόγραμμα. Για παράδειγμα, γράψτε πρώτα τη main() και ελέγξτε την ορθή λειτουργία της. Ελέγξτε την ορθή λειτουργία του μενού επιλογής.

Για κάθε προστιθέμενη νέα συνάρτηση, δώστε το πρωτότυπο της συνάρτησης. Σε πρώτη φάση, μπορείτε για σώμα να βάλετε μόνο μια printf που θα τυπώνει ένα μήνυμα με το όνομα της συνάρτησης. Αυτό θα σας βοηθήσει να ελέγξετε πρώτα τη ροή ελέγχου. Σε επόμενο βήμα, προσθέστε το σώμα της.

Σύμφωνα με τα παραπάνω, το πρόγραμμά μας μπορεί να αναπτυχθεί σε 7 διαδοχικές επαναλήψεις, όπου η κάθε μια θα προσθέτει νέα λειτουργικότητα σε αυτό. Οι διαδοχικές αυτές επαναλήψεις μπορεί να είναι:

Βήμα 1: Ορισμός της κύριας ροής του προγράμματος.

Βήμα 2: Υλοποίηση των δύο λειτουργιών της εισαγωγής λέξεων και της εμφάνισης τους (λειτουργίες 1 και 2).

Βήμα 3: Υλοποίηση της αύξουσας και φθίνουσας ταξινόμησης (λειτουργίες 3 και 4).

Βήμα 4: Υλοποίηση της υπηρεσίας παροχής στατιστικών στοιχείων που σχετίζονται με το μήκος λέξης (λειτουργία 5).

Βήμα 5: Υλοποίηση της υπηρεσίας παροχής στατιστικών στοιχείων που σχετίζονται με την εμφάνιση δεδομένου χαρακτήρα στις λέξεις του πίνακα (λειτουργία 6).

Βήμα 6: Υλοποίηση της υπηρεσίας αναζήτησης λέξης και εισαγωγής νέας λέξης στον πίνακα (λειτουργία 7).

Τελικό Βήμα: Γενικές βελτιώσεις, παροχή πρόσθετων υπηρεσιών.

11.4.4 Οδηγίες

Για να ακολουθήσετε την αυξητική ανάπτυξη, σας δίνουμε οδηγίες και βοήθεια για κάθε ένα από τα παραπάνω βήματα.

Βήμα 1. Ορισμός της κύριας ροής του προγράμματος

Στο βήμα αυτό, θα πρέπει να ορίσετε την κύρια ροή του προγράμματος. Σας συνιστώ να ακολουθήσετε τη λογική του μενού. Σύμφωνα με αυτή, το σύστημα θα δείνει στο χρήστη μια λίστα με τις διαθέσιμες υπηρεσίες που μπορεί να προσφέρει στα πλαίσια του συγκεκριμένου προγράμματος και θα ζητά από το χρήστη να δώσει την επιλογή του. Στη συνέχεια, θα παίρνει την επιλογή και θα επιλέγει την εκτέλεση των κατάλληλων ενεργειών οι οποίες θα πρέπει να είναι δομημένες με τρόπο που το σύστημα να παρέχει τελικά την ζητούμενη από το χρήστη υπηρεσία, π.χ. εμφάνιση λίστας λέξεων. Στη συνέχεια, θα εμφανίζει πάλι το μενού έως ότου ο χρήστης αποφασίσει τον τερματισμό του προγράμματος.

Η παραπάνω λογική θα πρέπει να ενσωματωθεί στον κώδικα της συνάρτησης main. Θα εφαρμόσετε βέβαια την αρχή της αφαιρετικότητας στις διαδικασίες σύμφωνα με την οποία στην πρώτη αυτή φάση δε θα ασχοληθείτε με το σώμα μιας διεργασίας αλλά κύρια με το όνομα που θα πρέπει να της αποδώσετε. Η επιλογή κατάλληλων ονομάτων για τις διεργασίες τις οποίες το πρόγραμμά σας περιγράφει αποτελεί το βασικότερο βήμα για την ανάπτυξη ενός ευανάγνωστου και επεκτάσιμου προγράμματος.

Εντοπίστε τις βασικές αυτές διεργασίες και αποδώστε τα κατάλληλα ονόματα με στόχο πάντα την αναγνωσημότητα της συνάρτησης main.

Δηλώστε μία συνάρτηση για κάθε μία από τις διεργασίες που εντοπίσατε.

Φροντίστε ώστε το όνομα της κάθε συνάρτησης να δηλώνει την εκτελούμενη διεργασία. Για παράδειγμα, η συνάρτηση που αναπαριστά τη διεργασία εισαγωγής λέξεων μπορεί να είναι η get_words().

Είναι προφανές πως προτάσεις όπως οι παρακάτω βελτιώνουν την αναγνωσιμότητα του κώδικα κάνοντας τα σχόλια περιττά.

```
while((selected_operation = menu())!=TERMINATE){
:
}
```

```
switch(selected_operation) {
    case GET_WORDS :
            get_words();
            break;
    case INCREMENTAL_SORT :
        :
}
```

Σχόλια για την πρόταση while

Το σύστημα θα υπολογίσει την τιμή της έκφρασης

```
(selected_operation = menu()) !=TERMINATE
```

και, αν αυτή είναι αληθής (κάθε διάφορη του 0 τιμή για τη C), θα εκτελέσει το σώμα του βρόχου επανάληψης.

Διαφορετικά, αν η τιμή είναι ψευδής (0 για τη C) θα τερματίσει το βρόχο επανάληψης και θα οδηγηθεί στην εκτέλεση της πρόταση που ακολουθεί το σώμα της `while`.

Για τον υπολογισμό της τιμής της έκφρασης

```
(selected_operation = menu()) !=TERMINATE
```

το σύστημα ενεργεί ως ακολούθως:

1. Καλεί τη συνάρτηση `menu`. Ενέργεια που προκύπτει από την έκφραση κλήσης συνάρτησης `menu()`.
2. Αποδίδει την επιστρεφόμενη τιμή της συνάρτησης στη μεταβλητή `selected_operation`. Ενέργεια που προκύπτει από την εφαρμογή του τελεστή ανάθεσης =.
3. Ελέγχει αν η τιμή της έκφρασης

   `(selected_operation = menu())`

   είναι διάφορη από την τιμή της σταθεράς `TERMINATE`. Ενέργεια που προκύπτει από την εφαρμογή του τελεστή ελέγχου ανισότητας !=.

Ταυτόχρονα με την εφαρμογή της αφαιρετικότητας στις διεργασίες, θα πρέπει να εφαρμόζετε και την αφαιρετικότητα στα δεδομένα. Ο εντοπισμός του κατάλληλου τύπου για τα δεδομένα του προβλήματος και η απόδοση κατάλληλων ονομάτων στις μεταβλητές θα πρέπει να είναι βασικός στόχος σας.

Η μεταβλητή που αναπαριστά την επιλογή του χρήστη είναι τύπου ακεραία ή απαριθμητικού όχι όμως πραγματικός αριθμός. Το δε όνομά της μπορεί να είναι choice, user_selection, selected_operation όχι όμως i, xyz, menou λέξεις που δεν αποδίδουν τη σημασία της αναπαριστώμενης πληροφορίας.

Προχωρήστε στην κατασκευή της συνάρτησης main και της συνάρτησης που υποστηρίζει τη διαδικασία επιλογής υπηρεσίας από το χρήστη.

- Δώστε μόνο δήλωση και ορισμό χωρίς σώμα για τις υπόλοιπες συναρτήσεις. Κάτι που βοηθά στη φάση αυτή είναι να ορίζετε ως σώμα της συνάρτησης μια printf που θα τυπώνει το όνομα της συνάρτησης. Αυτό θα σας βοηθήσει να ελέγξετε την ορθότητα της κύριας ροής του προγράμματός σας.

Βήμα 2. Υλοποίηση των δύο λειτουργιών της εισαγωγής λέξεων και της εμφάνισής τους

Προχωρήστε στο δεύτερο αυτό βήμα αφού έχετε πρώτα ολοκληρώσει τη συγγραφή του κώδικα που απαιτεί το βήμα 1 και έχετε μετεγλωττίσει και ελέγξει την ορθότητα αυτών που έχετε γράψει. Είναι ανώφελο να προχωρήσετε, αν δεν έχετε ολοκληρώσει το βήμα 1, απλά θα κάνετε την ολοκλήρωση του πιο δύσκολη.

Στο βήμα αυτό επιλέξαμε δύο υπηρεσίες μόνο και μόνο γιατί η δεύτερη είναι πολύ απλή. Παρόλα αυτά, σε πρώτη φάση θα πρέπει να ασχοληθείτε με την υπηρεσία της εισαγωγής λέξεων. Στη συνέχεια, αφού την ολοκληρώσετε και την ελέγξετε θα πρέπει να προχωρήσετε με την υπηρεσία της εμφάνισης.

Καθώς η σωστή οργάνωση των λέξεων θα αποτελέσει τη βάση για ένα ευέλικτο πρόγραμμα, χρησιμοποιήστε πίνακα δύο διαστάσεων για την αποθήκευση των λέξεων.

Σε πρώτη φάση και για να διευκολυνθεί η διαδικασία ανάπτυξης (κυρίως αυτή του ελέγχου του προγράμματός σας), ορίστε μικρό αριθμό λέξεων, π.χ. 4-5.

Ένας απλός τρόπος εισαγωγής μιας λέξης είναι με χρήση της scanf. Στην περίπτωση εισαγωγής συγκεκριμένου αριθμού λέξεων, θα πρέπει να επιλέξετε τον κατάλληλο βρόχο επανάληψης.

- Ολοκληρώστε τη συνάρτηση get_words.
- Ενσωματώστε τη στο πρόγραμμά σας και ελέγξτε τη λειτουργία της.
- Προχωρήστε στον ορισμό της συνάρτησης εμφάνισης.
- Βελτιώστε τη διαδικασία ελέγχου του προγράμματός σας. Στο σημείο αυτό, μπορείτε να χρησιμοποιήσετε το παρακάτω τέχνασμα για να βελτιώσετε τη διαδικασία ελέγχου του προγράμματός σας.
- Τροποποιήστε λίγο τη ροή του προγράμματός σας ώστε να εκτελεί την συνάρτηση εισαγωγής λέξεων πρίν την εμφάνιση του μενού.
- Δημιουργήστε ένα αρχείο txt με μια λίστα λέξεων και χρησιμοποιήστε την τεχνική της ανακατεύθυνσης (redirection) κατά την εκτέλεση του προγράμματος σας ώστε αυτό να παίρνει τις λέξεις από το αρχείο σας. Αν το πρό-

γραμμά σας είναι το askisi4 και το αχείο με τις λέξεις το lexeis.txt, η εντολή γραμμής διαταγής θα πρέπει να είναι

```
askisi4 < lexeis.txt
```

Προχωρήστε στο επόμενο βήμα μόνο όταν θα έχετε ολοκληρώσει και ελέγξει τον κώδικα που προσθέσατε στα πλαίσια του βήματος αυτού.

Βήμα 3. Υλοποίηση της αύξουσας ταξινόμησης

Για την ταξινόμηση υπάρχουν διάφοροι αλγόριθμοι. Μπορείτε να επιλέξετε έναν από αυτούς και να τον υλοποιήσετε φτιάχνοντας μία συνάρτηση. Πριν προχωρήσετε όμως προς την κατεύθυνση αυτή, θα πρέπει να εξετάσετε αν υπάρχει συνάρτηση της βασικής βιβλιοθήκης που εκτελεί τη λειτουργία της ταξινόμησης. Αν όχι, ίσως υπάρχει βιβλιοθήκη τρίτου κατασκευαστή (ή δική σας) που περιέχει τη ζητούμενη συνάρτηση.

Αν ανατρέξετε στην τεκμηρίωση της βασικής βιβλιοθήκης της C θα βρείτε τη συνάρτηση qsort της οποίας μια περιληπτική τεκμηρίωση παρατίθεται στη συνέχεια.

11.4.5 Συνάρτηση qsort

```
#include <stdlib.h>
#include <search.h>

void qsort(void *base, size_t num,size_t width,
           int (*comp)(const void *, const void *));
```

όπου *size_t* είναι ο απρόσημος ακέραιος τύπος που παράγει ο τελεστής sizeof.

Θυμίζουμε πως ο τελεστής ***sizeof*** εφαρμοζόμενος σε ένα τελεστέο (έκφραση, τύπο) δίνει το μέγεθος του σε bytes. Σε περίπτωση έκφρασης δίνει τον αριθμό των bytes που καταλαμβάνει η τιμή της έκφρασης.

Η *qsort* υλοποιεί ένα quick sort αλγόριθμο για να ταξινομήσει ένα πίνακα base, με αριθμό στοιχείων num και μέγεθος στοιχείου width. Για να μπορεί η συνάρτηση να ταξινομήσει διαφόρων τύπων αντικείμενα, όπως, για παράδειγμα, αριθμούς και αλφαριθμητικά, έχει υλοποιηθεί με τρόπο που να εκτελεί την ταξινόμηση κάνοντας τις απαραίτητες συγκρίσεις μεταξύ των στοιχείων χρησιμοποιώντας τη συνάρτηση comp. Τη συνάρτηση αυτή θα πρέπει να ορίσει ο χρήστης σύμφωνα με ορισμένες συμβάσεις που ορίζει το πρότυπο. Οι συμβάσεις αυτές ορίζουν τι πρέπει να δέχεται η συνάρτηση και τι να επιστρέφει.