

ΣΕΤ ΑΣΚΗΣΕΩΝ 2**ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020**

Προθεσμία: 8/12/2019, 23:59

Διαβάστε πριν ξεκινήσετε

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

Αποφασίστε τι μεταβλητές θα χρειαστείτε, τι ονόματα θα τους δώσετε, αν χρειάζονται σταθερές κι αν ναι για ποιες ποσότητες, τι δομές ελέγχου θα χρησιμοποιήσετε για κάθε λειτουργία, αν και ποιες επιπλέον συναρτήσεις χρειάζονται. Κάθε φορά που ολοκληρώνετε ένα στάδιο, επιβεβαιώστε ότι το πρόγραμμά σας λειτουργεί σωστά, πριν προχωρήσετε παρακάτω.

Αν χρειάζεστε διευκρινίσεις ή έχετε προβλήματα, στείλτε μήνυμα στο χώρο συζητήσεων στο e-class. Προσοχή: δεν επιτρέπεται η δημοσίευση κώδικα στο eclass.

Η εργασία αυτή μπορεί να γίνει σε ομάδες μέχρι και 2 ατόμων. Δε χρειάζεται να είστε ομάδα με το ίδιο άτομο που είστε στο εργαστήριο ή που κάνατε την προηγούμενη εργασία. Μπορείτε να συζητάτε τις ασκήσεις με συμφοιτητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιονδήποτε τρόπο.

Αν δουλέψετε σε ομάδα, τότε πριν ξεκινήσετε την εργασία συνδεθείτε στο Autolab, επιλέξτε το hw1 και κατασκευάστε μια ομάδα μέσω της επιλογής Group Options. Μην αφήσετε αυτή τη διαδικασία για τελευταία στιγμή!

Ξεκινήστε νωρίς! Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ότι περιμένετε.

Εκπρόθεσμες υποβολές των ασκήσεων δε γίνονται δεκτές.

Οι ασκήσεις σας θα βαθμολογηθούν στα παρακάτω (χωρίς ιδιαίτερη σειρά):

- Ορθότητα υπολογισμών
- Σωστή χρήση δεικτών, συναρτήσεων και πινάκων
- Αποτελεσματική χρήση κατάλληλων δομών, μεταβλητών, σταθερών κτλ.
- Γενική μορφοποίηση προγράμματος (στοίχιση, ονόματα μεταβλητών και σταθερών, κτλ.)
- Συμμόρφωση με τις προδιαγραφές
- Αποτελεσματικά σχόλια

Απαγορεύεται αυστηρά η χρήση goto, η χρήση gets και η χρήση καθολικών μεταβλητών.

Μπορείτε να υποθέσετε ότι η είσοδος από το πληκτρολόγιο θα δίνεται πάντα στη σωστή μορφή.

Διάβασμα κειμένου, αποθήκευση του στη μνήμη και επεξεργασία του

Σε αυτή την εργασία θα γράψετε ένα πρόγραμμα το οποίο διαβάζει ένα κείμενο λέξη-λέξη και:

1. Αποθηκεύει κάθε λέξη του κειμένου σε ένα πίνακα χαρακτήρων δύο διαστάσεων. Εάν μία λέξη εμφανίζεται περισσότερες από μία φορές στο κείμενο, αυτή αποθηκεύεται μία φορά στο συγκεκριμένο πίνακα.
2. Σε ένα ξεχωριστό πίνακα δεικτών δύο διαστάσεων αποθηκεύει τη σειρά με την οποία εμφανίζονται οι λέξεις μέσα στο κείμενο.

Συνοπτικά, οι λειτουργίες που προδιαγράφει η συγκεκριμένη εργασία είναι οι εξής:

1. Εκτύπωση του κειμένου, χρησιμοποιώντας συγκεκριμένους κανόνες ως προς την τελική του μορφοποίηση.
2. Δημιουργία ιστογράμματος. Για κάθε λέξη υπολογίζεται το πλήθος εμφανίσεων της στο κείμενο.
3. Υπολογισμός του χώρου που καταλαμβάνεται στη μνήμη στις παρακάτω δύο εναλλακτικές περιπτώσεις αποθήκευσης της πληροφορίας:
 - ο Κάθε λέξη του κειμένου αποθηκεύεται μόνο μία φορά στον πίνακα χαρακτήρων. Διατηρούμε παράλληλα ένα πίνακα δεικτών που προσδιορίζει τη σειρά των λέξεων μέσα στο κείμενο.
 - ο Το κείμενο αποθηκεύεται στο σύνολο του λέξη προς λέξη σε ένα πίνακα χαρακτήρων
4. Αντικατάσταση κάποιων από τις λέξεις του κειμένου με τα συνώνυμα τους. Η αντιστοιχία λέξης-συνώνυμης λέξης παρέχεται από ένα πίνακα χαρακτήρων δύο διαστάσεων που σας δίνεται έτοιμος.

Κώδικας που παρέχεται έτοιμος

Για την εργασία σας παρέχεται η στατική βιβλιοθήκη `libhw2.a`, το αρχείο `hw2.h`, το οποίο δεν επιτρέπεται να αλλάξετε και ο σκελετός του αρχείου `hw2.c` στο οποίο καλείστε να συμπληρώσετε τον κώδικα σας. Ο κώδικας που θα γράψετε θα πρέπει να περιέχεται εντός της συνάρτησης `hw2_main`. Η συνάρτηση `main` σας δίνεται έτοιμη και δεν πρέπει να αλλαχτεί.

Η βιβλιοθήκη `libhw2.a` περιέχει τις υλοποιήσεις των συναρτήσεων:

- `init_stack_size`: Διαμορφώνει το μέγεθος του `stack`, ώστε η εργασία να λειτουργεί και για μεγάλα αρχεία κειμένου. Η συνάρτηση καλείται μέσα στη συνάρτηση `main`.
- `read_synonyms`: Διαβάζει από ένα αρχείο τις αντιστοιχίες λέξεων και συνωνύμων. Περισσότερες πληροφορίες για τη συνάρτηση θα βρείτε στο στάδιο 4.

Για να μεταγλωττίσετε το πρόγραμμά σας, βεβαιωθείτε ότι το αρχείο `libhw2.a` βρίσκεται στον ίδιο κατάλογο με τα `hw2.c`, `hw2.h` και γράψτε:

```
gcc -Wall -g hw2.c -o hw2 -lhw2 -L.
```

Στάδιο 0 - Διάβασμα ορισμάτων/επιλογών εκτέλεσης από τη γραμμή εντολών

Το πρόγραμμα που θα γράψετε δέχεται από τη γραμμή εντολών τα ορίσματα με τα οποία θα εκτελεστεί και από την καθιερωμένη είσοδο (με ανακατεύθυνση σε αρχείο) το κείμενο προς επεξεργασία. Οι έγκυρες επιλογές που μπορεί να δοθούν από την γραμμή εντολών είναι οι εξής:

- `-p`: Εκτυπώνει το τελικό κείμενο.
- `-h`: Υπολογίζει το ιστόγραμμα των λέξεων του κειμένου και το εκτυπώνει στην καθιερωμένη έξοδο.
- `-r <filename>`: Εντοπίζει κάθε λέξη του κειμένου που περιέχεται στον πίνακα των συνώνυμων λέξεων και διατηρεί ένα δείκτη προς τη συνώνυμη της. Απαιτεί ως επιπλέον παράμετρο το όνομα αρχείου `<filename>` από το οποίο θα διαβαστούν οι συνώνυμες λέξεις. Το `<filename>` ακολουθεί υποχρεωτικά την παράμετρο `-r`.

Οι επιλογές μπορεί να δοθούν με οποιαδήποτε σειρά, αλλά η σειρά εκτέλεσης των αντίστοιχων λειτουργιών (ανεξάρτητα από την σειρά με την οποία δίνονται από τον χρήστη στην γραμμή εντολών) είναι η εξής: `-h` → `-r` → `-p`

Για παράδειγμα, αν ο χρήστης δώσει τις επιλογές `-p -h`, τότε πρώτα θα εκτελεστεί η λειτουργία που αντιστοιχεί στην επιλογή `-h` και μετά η λειτουργία που αντιστοιχεί στην επιλογή `-p`.

Κατά την εκτέλεση του προγράμματος θα πρέπει να παρέχεται μία ή περισσότερες από τις παραπάνω επιλογές στη γραμμή εντολών. Εάν δεν παρέχεται καμία επιλογή ή παρέχεται έστω και μία μη-έγκυρη επιλογή, εκτυπώνεται το μήνυμα "Incorrect command-line arguments!" ακολουθούμενο από χαρακτήρα αλλαγής γραμμής και το πρόγραμμα τερματίζει.

Στάδιο 1 - Διάβασμα του κειμένου από την καθιερωμένη είσοδο

Στη συνάρτηση `hw2_main` ορίστε τους παρακάτω πίνακες:

1. Ένα διδιάστατο πίνακα χαρακτήρων με αριθμό γραμμών `MAX_WORDS` και αριθμό στηλών `MAX_WORD_LEN`. Αρχικοποιήστε τον πίνακα ώστε να περιέχει σε όλες τις θέσεις του τον χαρακτήρα `'\0'`.
2. Ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό γραμμών `MAX_WORDS` και αριθμό στηλών 2. Αρχικοποιήστε τον πίνακα ώστε να περιέχει σε όλες τις θέσεις του την τιμή `NULL`.

Γράψτε μία συνάρτηση η οποία παίρνει ως παραμέτρους ένα διδιάστατο πίνακα χαρακτήρων με αριθμό στηλών `MAX_WORD_LEN` και ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό στηλών 2 και έχει τύπο επιστροφής `void`.

Η συνάρτηση διαβάζει ένα κείμενο λέξη-λέξη από την καθιερωμένη είσοδο μέχρι ή να γεμίσει ο πίνακας ή να τελειώσει η είσοδος. Κάθε λέξη του κειμένου χωρίζεται από την προηγούμενη και την επόμενη με λευκούς χαρακτήρες.

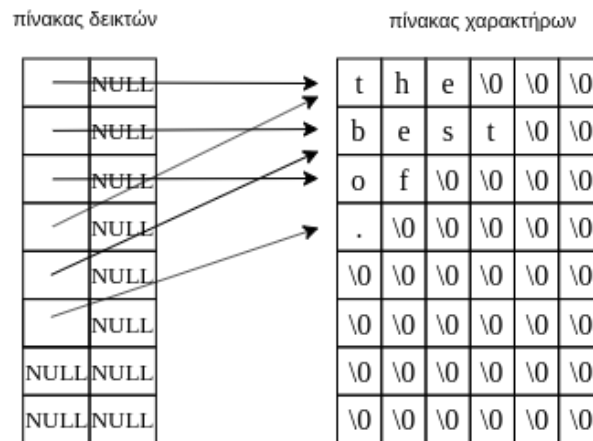
Για κάθε λέξη που διαβάζεται, η συνάρτηση κάνει τα εξής:

1. Μετατρέπει όλους τους κεφαλαίους χαρακτήρες της λέξης σε πεζούς. Για μετατροπές χαρακτήρων μπορείτε να χρησιμοποιήσετε κατάλληλες συναρτήσεις από το [ctype.h](#)
2. Εξετάζει εάν η λέξη υπάρχει στον πίνακα χαρακτήρων. Στον πίνακα χαρακτήρων καταχωρείται πάντα μία λέξη ανά γραμμή (ανεξάρτητα από το μήκος της). Θεωρήστε ότι ο πίνακας έχει επαρκές πλήθος στηλών για να αποθηκεύσει οποιαδήποτε λέξη του κειμένου.

- ο Εάν η λέξη υπάρχει στον πίνακα χαρακτήρων (έχει ήδη διαβαστεί μια φορά), εντοπίζει την διεύθυνση της στον πίνακα.
 - ο Εάν η λέξη δεν υπάρχει μέσα στον πίνακα χαρακτήρων, την προσθέτει στην πρώτη κενή γραμμή του πίνακα.
3. Προσθέτει μία νέα εγγραφή στον πίνακα δεικτών, ώστε η τελευταία μη κενή θέση της πρώτης στήλης του πίνακα δεικτών να δείχνει στην τελευταία λέξη που διαβάστηκε από το κείμενο και είναι αποθηκευμένη εντός του πίνακα χαρακτήρων (είτε πρόκειται για νέα εγγραφή, είτε για υφιστάμενη).

Σημείωση #1: Για να απλοποιηθεί η απαιτούμενη επεξεργασία, τα σημεία στίξης (. , ! ; ?) εμφανίζονται ως ξεχωριστές λέξεις μέσα στο κείμενο (μεσολαβούν λευκοί χαρακτήρες ανάμεσα σε ένα σημείο στίξης και την προηγούμενη καθώς και την επόμενη λέξη).

Για παράδειγμα, για το κείμενο "The best of the best ." ο πίνακας χαρακτήρων και ο πίνακας δεικτών διαμορφώνονται όπως στο ακόλουθο σχήμα.



Γράψτε κώδικα στη `hw2_main` ο οποίος διαβάζει ένα κείμενο λέξη-λέξη από την καθιερωμένη είσοδο, χρησιμοποιώντας τη συνάρτηση που μόλις γράψατε. Η συνάρτηση λαμβάνει ως παραμέτρους κατά την κλήση της το διδιάστατο πίνακα χαρακτήρων και το διδιάστατο πίνακα δεικτών που ορίσατε σε αυτό το στάδιο.

Σε αυτή τη φάση δεν γίνεται χρήση της δεύτερης στήλης του πίνακα δεικτών, τα στοιχεία της οποίας απλά παραμένουν αρχικοποιημένα σε NULL.

Σημείωση #2

Όταν πληκτρολογείτε τις λέξεις του κειμένου (και δεν τις δίνετε από αρχείο μέσω ανακατεύθυνσης της συμβατικής εισόδου) τότε πρέπει να πληκτρολογήσετε `control-D` για να σηματοδοτήσετε το τέλος του κειμένου (εισόδου).

Στάδιο 2 - Εκτύπωση του κειμένου

Γράψτε μία συνάρτηση που λαμβάνει ως παραμέτρους ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό στηλών 2 και έναν ακέραιο `linelen` και έχει επιστρεφόμενο τύπο `void`. Η συνάρτηση πρέπει να εκτυπώνει στην καθιερωμένη έξοδο το κείμενο στο οποίο αναφέρεται ο πίνακας δεικτών, ως εξής:

- Εκτυπώνει ένα χαρακτήρα αλλαγής γραμμής πριν την πρώτη λέξη του κειμένου.
- Η πρώτη λέξη του κειμένου και κάθε λέξη που έπεται τελείας, θαυμαστικού ή (αγγλικού) ερωτηματικού ξεκινάει με κεφαλαίο γράμμα.

- Οι λέξεις και τα σημεία στίξης διαχωρίζονται μεταξύ τους με ένα χαρακτήρα κενό (space).
- Ως εξαίρεση στον παραπάνω κανόνα, αν η επόμενη λέξη που θα προστεθεί οδηγεί σε γραμμή μήκους μεγαλύτερου των `linelen` χαρακτήρων, αντί για κενό χαρακτήρα εκτυπώνεται χαρακτήρας αλλαγής γραμμής και ακολουθεί η επόμενη λέξη (σε νέα γραμμή). Μπορείτε να υποθέσετε ότι το `linelen` θα είναι πάντα μεγαλύτερο ή ίσο του `MAX_WORD_LEN`.

Η διαδικασία επαναλαμβάνεται για ολόκληρο το κείμενο στο οποίο αναφέρεται ο πίνακας, έτσι ώστε όλες οι γραμμές να έχουν τη μέγιστη δυνατή πληρότητα λέξεων (χωρίς αυτές να διασπώνται), αλλά καμία γραμμή να μην ξεπερνά τους `linelen` χαρακτήρες. Στο τέλος εκτυπώνεται χαρακτήρας αλλαγής γραμμής.

Γράψτε κώδικα στη `hw2_main` ο οποίος, εφόσον ο χρήστης δώσει στην γραμμή εντολών την παράμετρο `-p`, καλεί την παραπάνω συνάρτηση με παραμέτρους τον πίνακα δεικτών που ορίστηκε / συμπληρώθηκε στο προηγούμενο στάδιο και τον ακέραιο `MAX_LINE_LEN` που προσδιορίζεται στο αρχείο `hw2.h`.

Στάδιο 3 - Υπολογισμός του ιστογράμματος

Στη συνάρτηση `hw2_main` ορίστε ένα πίνακα ακεραίων μεγέθους `MAX_WORDS` και αρχικοποιήστε τον ώστε να περιέχει σε όλες τις θέσεις του την τιμή 0. Κάθε θέση του πίνακα αντιστοιχεί στο πλήθος εμφανίσεων της λέξης η οποία βρίσκεται στην αντίστοιχη θέση του πίνακα χαρακτήρων.

Γράψτε μία συνάρτηση η οποία λαμβάνει τις εξής δύο παραμέτρους: α) ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό στηλών 2 και β) ένα μονοδιάστατο πίνακα ακεραίων. Η συνάρτηση έχει επιστρεφόμενο τύπο `void`.

Η συνάρτηση καλείται με παραμέτρους τον πίνακα δεικτών του σταδίου 1 και τον πίνακα ακεραίων που ορίσατε στο τρέχον στάδιο και αποθηκεύει στον πίνακα ακεραίων το πλήθος εμφανίσεων κάθε λέξης μέσα στο κείμενο.

Γράψτε μία άλλη συνάρτηση η οποία λαμβάνει ως παραμέτρους ένα διδιάστατο πίνακα χαρακτήρων με αριθμό στηλών `MAX_WORD_LEN` και ένα πίνακα ακεραίων. Η συνάρτηση καλείται με ορίσματα τον πίνακα χαρακτήρων που ορίστηκε στο στάδιο 1 και τον πίνακα ακεραίων που ορίσατε στο τρέχον στάδιο και δεν επιστρέφει κάτι. Η συνάρτηση εκτυπώνει στην έξοδο το ιστόγραμμα ως εξής:

Εκτυπώνει ένα χαρακτήρα αλλαγής γραμμής.

Διατρέχει τον πίνακα χαρακτήρων και για κάθε λέξη εκτυπώνει τη λέξη ώστε να καταλαμβάνει `MAX_WORD_LEN` θέσεις και να είναι στοιχισμένη δεξιά, άνω-κάτω τελεία, κενό και το πλήθος των εμφανίσεων της ως εξής:

- Εκτυπώνει τόσους χαρακτήρες '\$', όσες χιλιάδες φορές εμφανίζεται η λέξη στο κείμενο.
- Εφόσον αφαιρέσουμε τις χιλιάδες που εκτυπώθηκαν παραπάνω, εκτυπώνει τόσους χαρακτήρες '#', όσες εκατοντάδες φορές εμφανίζεται η λέξη στο κείμενο.
- Εφόσον αφαιρέσουμε τις χιλιάδες και τις εκατοντάδες που εκτυπώθηκαν παραπάνω, εκτυπώνει τόσους χαρακτήρες '@', όσες δεκάδες φορές εμφανίζεται η λέξη στο κείμενο.
- Εφόσον αφαιρέσουμε τις χιλιάδες, τις εκατοντάδες και τις δεκάδες που εκτυπώθηκαν παραπάνω, εκτυπώνει τόσους χαρακτήρες '*', όσες φορές εμφανίζεται η λέξη στο κείμενο.

Μετά από αυτή την "κωδικοποιημένη" εκτύπωση του πλήθους εμφανίσεων κάθε λέξης ακολουθεί χαρακτήρας αλλαγής γραμμής.

Για παράδειγμα εάν η λέξη "the" εμφανίζεται 2146 φορές θα εκτυπωθεί:

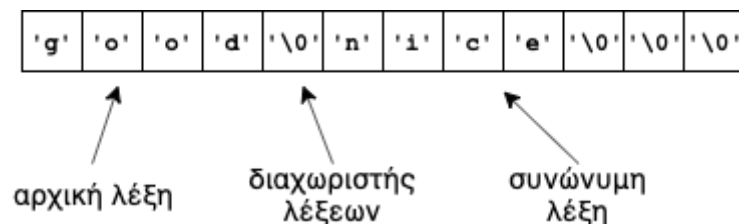
the: \$\$#@***

Αφού υλοποιήσετε τα παραπάνω συμπληρώστε κατάλληλα το τμήμα του κώδικα σας το οποίο ελέγχει αν δόθηκε από την γραμμή εντολών η παράμετρος -h και προβαίνει στις αντίστοιχες ενέργειες.

Στάδιο 4 - Εντοπισμός συνώνυμων λέξεων

Σε αυτό το στάδιο θα χρησιμοποιήσετε την έτοιμη συνάρτηση `read_synonyms` με πρώτο όρισμα το όνομα ενός αρχείου και δεύτερο όρισμα ένα διδιάστατο πίνακα χαρακτήρων στον οποίο θα αποθηκευτεί πληροφορία συνώνυμων λέξεων. Σε περίπτωση επιτυχίας, η συνάρτηση επιστρέφει 1, διαφορετικά (π.χ. αν το όνομα αρχείου είναι λανθασμένο) επιστρέφει 0.

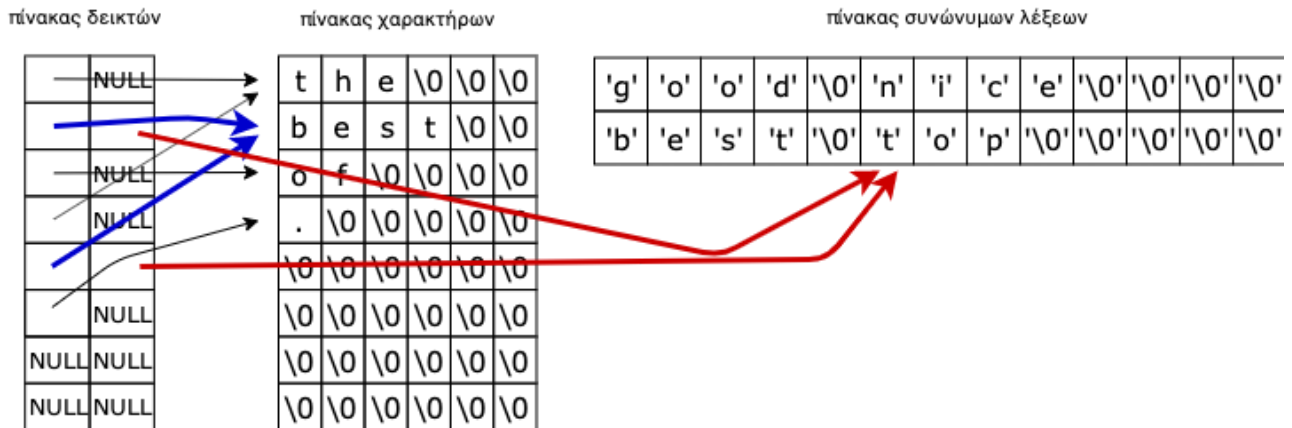
Η συνάρτηση υποθέτει ότι το αρχείο περιέχει σε κάθε γραμμή του δύο λέξεις με κενό ανάμεσά τους, εκ των οποίων η δεύτερη είναι συνώνυμη της πρώτης. Η συνάρτηση διαβάζει τα περιεχόμενα του αρχείου και αποθηκεύει σε κάθε μία γραμμή του πίνακα ζεύγη λέξεων και συνωνύμων αυτών, όπως φαίνεται στο παρακάτω σχήμα. Σε κάθε εγγραφή η αρχική λέξη χωρίζεται από τη συνώνυμη της με ένα χαρακτήρα '\0' που χρησιμεύει ταυτόχρονα και ως τερματικό της αρχικής λέξης, ενώ η συνώνυμη λέξη ακολουθεί στις επόμενες θέσεις του πίνακα και τερματίζεται και αυτή με ένα χαρακτήρα '\0'. Οι όποιες γραμμές στο τέλος του πίνακα δεν περιέχουν πληροφορία, περιέχουν σε όλες τις θέσεις τους τον χαρακτήρα '\0'.



Δημιουργήστε ένα διδιάστατο πίνακα χαρακτήρων με πλήθος γραμμών `MAX_SYNONYMS` και πλήθος στηλών `MAX_WORD_LEN*2` ο οποίος θα χρησιμοποιηθεί για την αποθήκευση πληροφορίας συνωνύμων λέξεων.

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παραμέτρους ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό στηλών 2 και ένα διδιάστατο πίνακα χαρακτήρων που περιέχει τις συνώνυμες λέξεις και έχει επιστρεφόμενο τύπο `void`. Η συνάρτηση καλείται με παραμέτρους τον διδιάστατο πίνακα δεικτών που ορίσατε στο στάδιο 1 και τον πίνακα χαρακτήρων που περιέχει τις συνωνυμες λέξεις και ορίσατε παραπάνω.

Η συνάρτηση λειτουργεί ως εξής: Διατρέχοντας την πρώτη στήλη του πίνακα δεικτών αναζητεί κατά πόσο κάθε λέξη στην οποία δείχνουν οι δείκτες του πίνακα βρίσκεται στις αρχικές λέξεις του πίνακα συνωνύμων λέξεων. Εάν βρεθεί μία τέτοια λέξη, αποθηκεύεται η διεύθυνση της συνωνύμης λέξης στη δεύτερη στήλη της τρέχουσας γραμμής του πίνακα δεικτών, διαφορετικά παραμένει η αρχική τιμή `NULL`. Το παρακάτω σχήμα δείχνει πως διαμορφώνεται το προηγούμενο παράδειγμα εάν για τη λέξη `best` το συνώνυμο είναι `top`.



Γράψτε κώδικα ο οποίος στην περίπτωση που ο χρήστης προσδιορίσει στη γραμμή εντολών την επιλογή `-r` ακολουθούμενη από το όνομα ενός αρχείου κάνει τα εξής:

1. Καλεί τη συνάρτηση `read_synonyms` δίνοντας ως πρώτη παράμετρο το όνομα του αρχείου και ως δεύτερη παράμετρο τον πίνακα των συνώνυμων λέξεων.
2. Αν η κλήση είναι επιτυχής, καλεί τη συνάρτηση που ορίσατε στο τρέχον βήμα, ώστε να ενημερώνει τον πίνακα δεικτών με τις συνώνυμες λέξεις που εμφανίζονται στο κείμενο.

Ενημερώστε τη συνάρτηση εκτύπωσης του σταδίου 2, ώστε εάν μία λέξη διαθέτει συνώνυμη λέξη στη δεύτερη στήλη του πίνακα δεικτών, τότε μετά την εκτύπωση της λέξης εκτυπώνεται η συνώνυμη αυτής ανάμεσα σε παρενθέσεις και ακολουθώντας τους κανόνες που περιγράφονται στο στάδιο 2 για τις αποστάσεις ανάμεσα στις λέξεις, για το μήκος της γραμμής και για τη χρήση κεφαλαίου γράμματος.

Στάδιο 5 (Προαιρετικό) - Υπολογισμός του κόστους αποθήκευσης

Θέλουμε να υπολογίσουμε το συνολικό κόστος αποθήκευσης της πληροφορίας (του αρχικού κειμένου χωρίς να ληφθεί υπόψη η πληροφορία συνωνύμων) για τους εξής δύο εναλλακτικούς τρόπους αποθήκευσης:

- A. Με τον τρόπο που αποθηκεύουμε την πληροφορία όπως περιγράφεται παραπάνω, δηλαδή κάθε λέξη εμφανίζεται μία μόνο φορά στον διδιάστατο πίνακα χαρακτήρων. Για κάθε λέξη διατηρούμε μία ή περισσότερες εγγραφές στον πίνακα δεικτών, ανάλογα με τον αριθμό εμφανίσεων της λέξης αυτής στο κείμενο. Σε αυτή την περίπτωση αθροίζουμε τα μήκη των λέξεων που αποθηκεύονται στον πίνακα χαρακτήρων και τον αριθμό των εγγραφών στον πίνακα δεικτών επί το μέγεθος ενός δείκτη.
- B. Εάν υποθετικά αποθηκεύαμε κάθε λέξη του κειμένου σε ξεχωριστή γραμμή του διδιάστατου πίνακα χαρακτήρων και κάθε λέξη εμφανιζόταν σε αυτόν όσες φορές εμφανίζεται και στο κείμενο. Σε αυτή την περίπτωση, το κόστος αποθήκευσης προκύπτει εάν αθροίσουμε τα μήκη των λέξεων που αποθηκεύονται στο διδιάστατο πίνακα.

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παραμέτρους ένα διδιάστατο πίνακα χαρακτήρων με αριθμό στηλών `MAX_WORD_LEN` και ένα διδιάστατο πίνακα δεικτών σε χαρακτήρα με αριθμό στηλών 2 και έχει επιστρεφόμενο τύπο `long int`. Η συνάρτηση επιστρέφει το μέγεθος που καταλαμβάνει το κείμενο εάν αποθηκευτεί με την υποθετική μέθοδο αποθήκευσης A.

Ομοίως, γράψτε μία άλλη συνάρτηση η οποία λαμβάνει τις ίδιες παραμέτρους και έχει τον ίδιο

επιστρεφόμενο τύπο με την προηγούμενη συνάρτηση και επιστρέφει το μέγεθος που καταλαμβάνει το κείμενο εάν αποθηκευτεί με τη μέθοδο αποθήκευσης B.

Γράψτε μία τρίτη συνάρτηση η οποία λαμβάνει τις ίδιες παραμέτρους με τις δύο προηγούμενες συναρτήσεις και έχει επιστρεφόμενο τύπο `double`. Η συνάρτηση καλείται με παραμέτρους τον πίνακα χαρακτήρων και τον πίνακα δεικτών που ορίστηκαν στο στάδιο 1 και εκτυπώνει το λόγο του μεγέθους της μνήμης που καταλαμβάνεται εάν επιλέξουμε ως μέθοδο αποθήκευσης τη μέθοδο A σε σχέση με τη μέθοδο B.

Εφόσον υλοποιήσετε τις παραπάνω συναρτήσεις, συμπληρώστε κατάλληλα το τμήμα του κώδικα σας που ελέγχει τα ορίσματα του προγράμματος, ώστε να υποστηρίζεται η επιπλέον επιλογή -s. Στην περίπτωση που δοθεί αυτή η επιλογή εκτυπώνεται ως τελευταία ενέργεια του προγράμματος χαρακτήρας αλλαγής γραμμής, ο παραπάνω λόγος με ακρίβεια 3 δεκαδικών ψηφίων και χαρακτήρας αλλαγής γραμμής.

Σκεφτείτε: Ποιος από τους δύο τρόπους αποθήκευσης καταλαμβάνει τον λιγότερο χώρο; Το μέγεθος ή/και το είδος του κειμένου επηρεάζουν το αποτέλεσμα; Μπορείτε να εξηγήσετε γιατί συμβαίνει αυτό;

Οδηγίες υποβολής εργασίας στο Autolab

1. Προσθέστε σε σχόλια στην αρχή του αρχείου `hw2.c` τα πλήρη ονόματα και ΑΕΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με αγγλικούς χαρακτήρες.
2. Κατασκευάστε ένα κατάλογο με όνομα `hw2submit` και αντιγράψτε μέσα σε αυτόν το `hw2.c`
3. Κατασκευάστε ένα αρχείο κειμένου με όνομα `team.txt` και προσθέστε σε αυτό τα πλήρη ονόματα και τα ΑΕΜ των μελών της ομάδας, ακόμη κι αν η ομάδα αποτελείται από ένα άτομο.
4. Κάντε δεξί κλικ στον κατάλογο `hw2submit` και επιλέξτε `Compress here as tar.gz`. Θα δημιουργηθεί ένα αρχείο με όνομα `hw2submit.tar.gz`.
5. Συνδεθείτε στο Autolab κι επιλέξτε το `hw2`.
 - Αν είστε ομάδα δύο ατόμων και δεν το έχετε ήδη κάνει, κατασκευάστε μια ομάδα μέσω της επιλογής `Group Options`.
6. Κάντε `Submit` το `hw2submit.tar.gz`.