

Προγραμματιστικές Ασκήσεις, Φυλλάδιο 4
Να ολοκληρωθούν μέχρι 25/1/12 στις 11:59 μ.μ.

Η σειρά ασκήσεων αυτή πρέπει να παραδοθεί μέχρι τις 25 Ιανουαρίου τα μεσάνυχτα. Εκτός από τις ερωτήσεις σε αυτό το φυλλάδιο για να πάρετε πλήρη βαθμό σε αυτή τη σειρά θα πρέπει να παραδώσετε και μία άσκηση στο εργαστήριο της Πέμπτης 19 Ιανουαρίου. Για την υλοποίηση των ασκήσεων θα πρέπει να χρησιμοποιήσετε κλάσεις και αντικείμενα. Στις κλάσεις δεν πρέπει να χρησιμοποιείτε public μεταβλητές. Εκτός από σταθερές, δεν πρέπει να χρησιμοποιήσετε καθολικές μεταβλητές (global variables). Τα προγράμματα σας πρέπει να εξασφαλίζουν ότι αποδεσμεύτηκε όλη η μνήμη η οποία είχε δεσμευτεί.

1. Υλοποιείτε ξανά την Ερώτηση 1 της Άσκησης 2, αλλά σπάσετε τον κώδικα σε πολλά αρχεία. Θα πρέπει να έχετε ένα **.h** και ένα **.cpp** αρχείο για κάθε κλάση που χρησιμοποιείτε, και να έχετε τουλάχιστον 3 κλάσεις (Car, Position, Dimension). Θα πρέπει επίσης να παραδώσετε και το makefile για το compiling των αρχείων που παραδώσατε. Η άσκηση θα κριθεί με βάση το αν έχει γίνει σωστά η διάσπαση του κώδικα σε αρχεία και αν γίνεται σωστά το compiling. Μια πρότυπη λύση για την ερώτηση (σε ένα αρχείο) είναι στη σελίδα του μαθήματος.
2. Η δεύτερη άσκηση θα δοθεί στο εργαστήριο της Πέμπτης 19 Ιανουαρίου.
3. Στην άσκηση αυτή θα υλοποιήσουμε μια πολύ απλή μηχανή αναζήτησης. Η άσκηση έχει δύο κομμάτια. Στο πρώτο θα κάνουμε ανάλυση των κειμένων που έχουμε, και στο δεύτερο θα επιτρέπουμε στον χρήστη να κάνει ερωτήματα. Για την άσκηση αυτή θα χρειαστεί να χρησιμοποιήσετε τις δομές της STL, και ειδικά την δομή map.
 - a. Στην ιστοσελίδα του μαθήματος θα βρείτε ένα zip αρχείο text.zip με N = 15 κείμενα από online reviews για προϊόντα. Τα αρχεία έχουν όνομα id.txt. Τα reviews 1-3 είναι για laptops, τα 4-6 για φωτογραφικές μηχανές, τα 7-9 για κινητά τηλέφωνα, τα 10-12 για mp3 players, και τα 13-15 για τηλεοράσεις. Το κάθε κείμενο είναι μία μόνο γραμμή. Σε αυτό το κομμάτι της ερώτησης πρέπει να διαβάσετε το κείμενο από το αρχείο, να το σπάσετε σε λέξεις (tokens), και θα μετατρέψετε τα κεφαλαία σε μικρά (lowercase). (Αν θέλετε να κάνετε τη μηχανή αναζήτησης σας καλύτερη αφαιρέστε και όλα τα σημεία στίξεως). Για κάθε κείμενο θα δημιουργήσετε ένα νέο αρχείο με όνομα id.tokens το οποίο θα αποθηκεύει για κάθε token την συχνότητα εμφανίσεων της λέξης μέσα στο κείμενο, όπου η συχνότητα είναι ο αριθμός των εμφανίσεων της λέξης, δια το συνολικό αριθμό λέξεων στο κείμενο. Το αποτέλεσμα για τα κείμενα της ερώτησης είναι στην ιστοσελίδα του μαθήματος στο αρχείο tokens.zip.
 - b. Στο δεύτερο βήμα θα διαβάσετε τα αρχεία id.tokens από το πρώτο βήμα (μπορείτε να χρησιμοποιήσετε τα αρχεία στη σελίδα του μαθήματος) και θα δημιουργήσετε ένα inverted index. Για κάθε λέξη θα κρατάτε μία λίστα με τα ids των κειμένων στα οποία εμφανίζεται αυτή η λέξη, και τη συχνότητα της λέξης στο αντίστοιχο κείμενο. Αφού φτιάξετε το inverted index θα πρέπει να απαντάτε τα ερωτήματα του χρήστη. Ένα ερώτημα είναι μια λίστα από λέξεις. Ένα κείμενο απαντά το ερώτημα αν περιέχει **όλες** τις λέξεις του ερωτήματος. Για ένα κείμενο που απαντά την ερώτηση υπολογίζουμε ένα σκορ που δείχνει πόσο σχετικό είναι το κείμενο ως προς την ερώτηση. Το σκορ είναι το άθροισμα των συχνοτήτων των λέξεων του ερωτήματος μέσα στο κείμενο. Ως απάντηση στο ερώτημα θα επιστρέφετε τα κείμενα (ids) που απαντάνε το ερώτημα και το σκορ τους, ταξινομημένα από το μεγαλύτερο προς το μικρότερο σκορ (ή ένα μήνυμα αν δεν υπάρχει κείμενο που απαντάει στο ερώτημα).

Η λύση σας θα πρέπει να είναι γενική ώστε να δουλεύει για οποιοδήποτε αριθμό N από κείμενα.

Bonus: +30% του βαθμού αν κάνετε την άσκηση σε Java.

ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

Κάντε turnin τα προγράμματα σας στο lab4@cs435. Για διευκόλυνση ονομάστε τα αρχεία με τον κώδικα σας **lab3_x_dddd.cpp** και τα αρχεία των αποτελεσμάτων **res3_x_dddd.txt** όπου dddd είναι ο Α.Μ. σας και x ο αριθμός της άσκησης.

π.χ. turnin lab3@cs435 lab3_1_dddd.cpp res3_1_dddd.txt lab3_3_dddd.cpp res3_3_dddd.txt

Στον κώδικα να αναγράφονται σε σχόλια το όνομα το login και ο ΑΜ σας.