Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
https://helios.ntua.gr/course/view.php?id=869

Προγραμματιστικές Τεχνικές

Άσκηση 10 Δυαδικά Δένδρα

Προθεσμία υποβολής στον grader: 8/5/2026

Υλοποιήστε την κλάση `lexicon` για την αναπαράσταση λεξιλογίων κειμένων με τη μορφή δυαδικών δένδρων αναζήτησης. Η κλάση θα πρέπει να υποστηρίζει τις παρακάτω λειτουργίες, τις οποίες πρέπει να υλοποίησετε:

```
class lexicon {
public:
  lexicon();
  ~lexicon();

  void insert(const string &s);
  int  lookup(const string &s) const;
  int  depth(const string &s);
  void replace(const string &s1, const string &s2);

  friend ostream & operator << (ostream &out, const lexicon &l);
};
```

Θεωρήστε δεδομένο ότι οι λέξεις που θα αποθηκεύονται σε ένα λεξικό δεν θα είναι κενές και θα περιέχουν μόνο μικρά λατινικά γράμματα. Κάθε λέξη θα αντιστοιχεί σε έναν κόμβο στο δένδρο. Ο κόμβος θα περιέχει τη λέξη και τη συχνότητα εμφάνισής της στο λεξικό (δηλαδή έναν μετρητή που θα θυμάται πόσες φορές έχει εισαχθεί αυτή η λέξη). Το αριστερό παιδί του κόμβου θα περιέχει αλφαβητικά (λεξικογραφικά) μικρότερες λέξεις, ενώ το δεξιό παιδί αλφαβητικά (λεξικογραφικά) μεγαλύτερες. Καμία λέξη δεν θα πρέπει να εμφανίζεται σε περισσότερους από έναν κόμβους στο δένδρο. Επίσης, το δένδρο θα πρέπει να είναι ένα απλό BST: δεν απαιτείται και δεν πρέπει να είναι ισοζυγισμένο (π.χ., AVL).

Η μέθοδος `insert(s)` θα εισάγει τη λέξη `s` στο δένδρο.

Η μέθοδος `lookup(s)` θα αναζητά τη λέξη `s` στο δένδρο και θα επιστρέφει τη συχνότητα εμφάνισής. Το αποτέλεσμα θα είναι 0 (μηδέν) αν η λέξη δεν υπάρχει στο δένδρο.

Η μέθοδος `depth(s)` θα αναζητά τη λέξη `s` στο δένδρο, όπως και η `lookup`. Αν δεν υπάρχει, θα επιστρέφει -1. Αν όμως υπάρχει, θα επιστρέφει το *βάθος* στο οποίο βρίσκεται ο κόμβος που την περιέχει στο δένδρο, δηλαδή το μήκος του μονοπατιού από τη ρίζα έως αυτόν τον κόμβο. Θεωρούμε ότι η ρίζα του δένδρου βρίσκεται σε βάθος 0 (μηδέν).

Η μέθοδος `replace(s1, s2)` Θα αντικαθιστά όλες τις εμφανίσεις της λέξης `s1` με ισάριθμες εμφανίσεις της λέξης `s2`. Αν η `s1` δεν υπάρχει στο δένδρο, τότε δε θα γίνεται τίποτα. Αν υπάρχει και το πλήθος εμφανίσεών της είναι $k > 0$, τότε η `s1` θα διαγράφεται και θα αναζητάται η λέξη `s2`. Αν αυτή δεν υπάρχει, θα εισάγεται με συχνότητα εμφάνισης k. Αν όμως υπάρχει, τότε η συχνότητα εμφάνισής της θα ενημερώνεται κατάλληλα (θα αυξάνει κατά k).

Κατά τη διαγραφή μίας λέξης από το λεξικό (που γίνεται έμμεσα με τη μέθοδο `replace`), αν διαγράφεται κόμβος που έχει δύο μη κενά παιδιά, τότε ο κόμβος που διαγράφεται αντικαθίσταται από αυτόν

που περιέχει την αμέσως μικρότερη λέξη. Αν διαγράφεται κόμβος που έχει ένα μη κενό παιδί, τότε αντικαθίσταται από το παιδί του.

Η εκτύπωση των λεξικών πρέπει να γίνεται σε αλφαβητική σειρά, με τις λέξεις να ακολουθούνται από τη συχνότητα εμφάνισής τους.

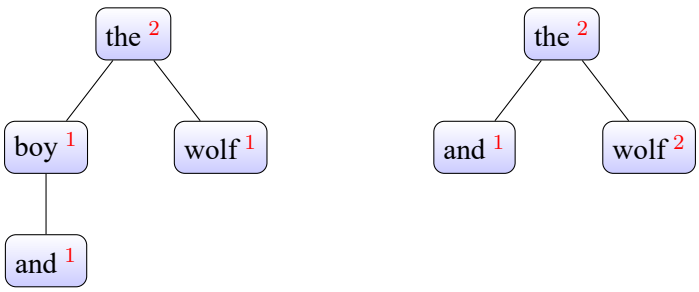
Μπορείτε να δοκιμάσετε την υλοποίησή σας με προγράμματα όπως το εξής:

```
int main() {
  lexicon l;
  l.insert("the");
  l.insert("boy");
  l.insert("and");
  l.insert("the");
  l.insert("wolf");
  cout << "The word 'the' is found " << l.lookup("the") << " time(s)" << endl;
  cout << "The word 'and' is found at depth " << l.depth("and") << endl;
  cout << l;
  l.replace("boy", "wolf");
  cout << "After replacement:\n";
  cout << l;
  cout << "Now the word 'and' is found at depth " << l.depth("and") << endl;
}
```

Η εκτέλεσή του θα πρέπει να εμφανίζει:

```
The word 'the' is found 2 time(s)
The word 'and' is found at depth 2
and 1
boy 1
the 2
wolf 1
After replacement:
and 1
the 2
wolf 2
Now the word 'and' is found at depth 1
```

Η μορφή του δένδρου πριν (αριστερά) και μετά (δεξιά) την κλήση της `replace` δίνεται παρακάτω.



Στο αρχείο που θα ανεβάσετε στον grader θα πρέπει να συμπεριλάβετε (μόνο) τη δήλωση της κλάσης `lexicon` και τις υλοποιήσεις των μεθόδων της.

Άσκηση 11 Ταινίες με τις καλύτερες αξιολογήσεις

Προθεσμία υποβολής στον grader: 8/5/2026

Αυτή η άσκηση, καθώς και η επόμενη, εστιάζει στη χρήση της βιβλιοθήκης STL, χωρίς την οποία η υλοποίηση των ζητούμενων λύσεων μπορεί να γίνει αρκετά πιο σύνθετη. Αντικείμενο των δύο αυτών ασκήσεων είναι η επεξεργασία των δεδομένων ταινιών και αξιολογήσεων που δίνονται στην είσοδο, με σκοπό τον υπολογισμό των ταινιών με τις καλύτερες αξιολογήσεις (σε αυτή την άσκηση) και των ειδών ταινιών (genres) με τον μεγαλύτερο αριθμό (πλήθος) αξιολογήσεων (στην επόμενη).

Σε αυτή την άσκηση, δίνεται στην πρώτη γραμμή της εισόδου ένας ακέραιος αριθμός K. Οι υπόλοιπες γραμμές της εισόδου περιέχουν δεδομένα αξιολόγησης ταινιών από θεατές. Σε κάθε τέτοια γραμμή δίνονται (με αυτή τη σειρά) ο τίτλος μιας ταινίας, ένα αναγνωριστικό του θεατή και η αξιολόγηση, χωρισμένα ανά δύο με ένα κενό διάστημα.

```
K
movie_title_1 viewer_1 rating_1
movie_title_2 viewer_2 rating_2
movie_title_3 viewer_3 rating_3
...
```

Να γράψετε ένα πρόγραμμα που να διαβάζει τα παραπάνω από την είσοδο και να εμφανίζει στην έξοδο τους τίτλους των K ταινιών με την καλύτερη μέση τιμή των αξιολογήσεών τους, με φθίνουσα κατάταξη (δηλαδή να εμφανίζεται πρώτη η ταινία με τη μεγαλύτερη αξιολόγηση). Η έξοδος πρέπει να αποτελείται από K γραμμές. Σε κάθε γραμμή πρέπει να εμφανίζεται ο τίτλος της ταινίας ακολουθούμενος από ένα κενό διάστημα και τη μέση τιμή των αξιολογήσεων αυτής:

```
movie_title_1 rating_1
movie_title_2 rating_2
...
movie_title_K rating_K
```

Θεωρήστε ότι για τα δεδομένα εισόδου και εξόδου θα ισχύουν τα ακόλουθα:

- Το πλήθος των αξιολογήσεων δεν δίνεται, δηλαδή πρέπει να διαβαστούν από την είσοδο όλες οι γραμμές μέχρι το τέλος αρχείου.
- Στον τίτλο των ταινιών και στα αναγνωριστικά των θεατών δε θα υπάρχουν κενά διαστήματα.
- Κάθε αξιολόγηση είναι ένας ακέραιος αριθμός και έχει αποδεκτές τιμές από 0 μέχρι και 100. Ωστόσο, δεν αποκλείεται να υπάρχει "θόρυβος" στα δεδομένα εισόδου, δηλαδή να υπάρχουν ακέραιες αριθμητικές τιμές εκτός του διαστήματος $[0, 100]$. Ο θόρυβος δεν πρέπει να λαμβάνεται υπόψη στους υπολογισμούς.
- Ο υπολογισμός των μέσων τιμών αξιολογήσεων γίνεται με απλή ακέραια διαίρεση του αθροίσματος, χωρίς στρογγυλοποίηση (π.χ., $100/3 = 33$, $150/7 = 21$, $170/9 = 18$, κ.ο.κ.).
- Σε περίπτωση ισοβαθμίας, πρέπει να εμφανίζεται πρώτη η ταινία που προηγείται αλφαβητικά.
- Η έξοδος του προγράμματος πρέπει να περιέχει K γραμμές, εκτός αν όλες οι ταινίες είναι λιγότερες από K, οπότε σε αυτή την περίπτωση θα περιέχει τόσες γραμμές όσες είναι όλες οι ταινίες.
- $1 \leq \mathsf{K} \leq 700$
- Το πλήθος των αξιολογήσεων δε θα υπερβαίνει το 100.000.

Παραδείγματα εισόδου και εξόδου μπορείτε να βρείτε στην εκφώνηση στον grader.

Άσκηση 12 Είδη ταινιών με τις περισσότερες αξιολογήσεις

Προθεσμία υποβολής στον grader: 8/5/2026

Στην πρώτη γραμμή της εισόδου δίνονται δύο ακέραιοι αριθμοί `N` και `G`. Σε κάθε μία από τις επόμενες `N` γραμμές δίνονται ο τίτλος μιας ταινίας και το είδος (genre) αυτής, με αυτή τη σειρά, χωρισμένα μεταξύ τους με ένα κενό διάστημα. Σε κάθε επόμενη γραμμή μέχρι το τέλος της εισόδου δίνονται, με αυτή τη σειρά, ο τίτλος μιας ταινίας, το αναγνωριστικό ενός θεατή και η αξιολόγησή του, χωρισμένα ανά δύο με ένα κενό διάστημα.

```
N G
movie_title_1 genre_1
movie_title_2 genre_2
...
movie_title_N genre_N
movie_title_1 viewer_1 rating_1
movie_title_2 viewer_2 rating_2
movie_title_3 viewer_3 rating_3
...
```

Να γράψετε ένα πρόγραμμα που να διαβάζει τα παραπάνω από την είσοδο και να εμφανίζει στην έξοδο τα G είδη ταινιών (genres), των οποίων οι ταινίες έχουν λάβει τις περισσότερες (σε πλήθος) αξιολογήσεις, με φθίνουσα κατάταξη (δηλαδή να εμφανίζεται πρώτο το είδος με τις περισσότερες αξιολογήσεις) ως ακολούθως:

```
genre_1 ratings_count_1
genre_2 ratings_count_2
...
genre_G ratings_count_G
```

Θεωρήστε ότι για τα δεδομένα εισόδου και εξόδου θα ισχύουν τα ακόλουθα:

- Η τιμή του N (ο αριθμός των ταινιών και ταξινομήσεων είδους που δίνονται αρχικά) είναι γνωστή.
- Το πλήθος των αξιολογήσεων δεν δίνεται, δηλαδή πρέπει να διαβαστούν από την είσοδο όλες οι γραμμές μέχρι το τέλος αρχείου.
- Κάθε ταινία αντιστοιχεί σε ένα μόνο είδος, είναι όμως πιθανό η ίδια αντιστοίχιση ταινίας σε είδος να δίνεται περισσότερες φορές.
- Στον τίτλο των ταινιών, στα ονόματα των ειδών ταινιών (genre) και στα αναγνωριστικά των θεατών δε θα υπάρχουν κενά διαστήματα.
- Κάθε αξιολόγηση είναι ένας ακέραιος αριθμός και έχει αποδεκτές τιμές από 0 μέχρι και 100. Ωστόσο, δεν αποκλείεται να υπάρχει "θόρυβος" στα δεδομένα εισόδου, δηλαδή να υπάρχουν ακέραιες αριθμητικές τιμές εκτός του διαστήματος $[0, 100]$. Ο θόρυβος δεν πρέπει να λαμβάνεται υπόψη στους υπολογισμούς.
- Αν μία αξιολόγηση αφορά ταινία της οποίας το είδος δεν είναι γνωστό, τότε επίσης θεωρείται "θόρυβος" και δεν πρέπει να λαμβάνεται υπόψη στους υπολογισμούς.
- Σε περίπτωση ισοβαθμίας, πρέπει να εμφανίζεται πρώτο το είδος που προηγείται αλφαβητικά.
- Η έξοδος του προγράμματος πρέπει να περιέχει `G` γραμμές, εκτός αν όλα τα είδη είναι λιγότερα από `G`, οπότε σε αυτή την περίπτωση θα περιέχει τόσες γραμμές όσες είναι όλα τα είδη.
- $1 \leq$ `N` ≤ 700 και $1 \leq$ `G` ≤ 30
- Το πλήθος των αξιολογήσεων δε θα υπερβαίνει το 100.000.

Παραδείγματα εισόδου και εξόδου μπορείτε να βρείτε στην εκφώνηση στον grader.

Υπόδειξη: Αξιοποιώντας τη βιβλιοθήκη STL, ίσως σας χρειαστεί μια αντιστοίχιση A μεταξύ τίτλου ταινίας και είδους. Επίσης, ίσως σας χρειαστεί μία αντιστοίχιση B του κάθε είδους με το πλήθος των αξιολογήσεων που το αφορά. Καθώς οι γραμμές της εισόδου με τις αξιολογήσεις δεν περιέχουν το είδος της ταινίας αλλά μόνο τον τίτλο της, η αντιστοίχιση B προϋποθέτει την ύπαρξη της αντιστοίχισης A.