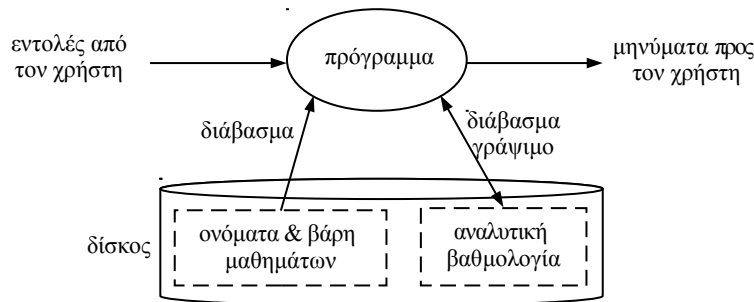


### Εργασία 1 – Διαχείριση αναλυτικής βαθμολογίας

Αναπτύξτε ένα πρόγραμμα για την διαχείριση της αναλυτικής βαθμολογίας ενός φοιτητή, τα δεδομένα της οποίας αποθηκεύονται σε αρχείο. Το πρόγραμμα πρέπει να υποστηρίζει τις εξής εντολές / λειτουργίες:

Λειτουργία	Δεδομένα	Περιγραφή
Προσθήκη	Μάθημα (char 64) Ημερομηνία (MM/YY) Βαθμός (NN.N)	Καταχώρηση εγγραφής και εκτύπωση επιβεβαίωσης (ή μηνύματος λάθους σε περίπτωση που κάποια τιμή είναι προβληματική ή ήδη υπάρχει εγγραφή για αυτό το μάθημα και την ημερομηνία εξέτασης).
Διαγραφή	Μάθημα (char 64) Ημερομηνία (MM/YY)	Διαγραφή εγγραφής και εκτύπωση επιβεβαίωσης (ή μηνύματος λάθους αν δεν υπάρχει εγγραφή για αυτό το μάθημα και την ημερομηνία εξέτασης).
Αναζήτηση	Μάθημα (char 64)	Εύρεση και εκτύπωση όλων των εγγραφών για αυτό το μάθημα.
Μέσος όρος	---	Εκτύπωση του αριθμού των μαθημάτων με βαθμό $\geq 5$ και του μέσου όρου με βάση το «βάρος» κάθε μαθήματος.

Το πρόγραμμα πρέπει να προσφέρει στον χρήστη κατάλληλο μενού με τις παραπάνω λειτουργίες, και να ελέγχει ότι τα ονόματα των μαθημάτων, οι ημερομηνίες εξέτασης και οι βαθμοί που δίνει ο χρήστης έχουν επιτρεπτές τιμές. Ένα μάθημα μπορεί να έχει πολλές εγγραφές με βαθμό  $< 5$  αλλά το πολύ μια με  $\geq 5$ . Τα δεδομένα πρέπει να αποθηκεύονται στο αρχείο σε μορφή ASCII έτσι ώστε να τα επαληθεύσετε εύκολα (π.χ., εκτυπώνοντας τα περιεχόμενα του αρχείου με more ή cat). Τα ονόματα και τα «βάρη» των μαθημάτων πρέπει να βρίσκονται σε ξεχωριστό αρχείο (ενδεικτικά, δείτε το αρχείο courses.txt στην ιστοσελίδα του μαθήματος, το οποίο μπορείτε να χρησιμοποιήσετε ως έχει). Η αλληλεπίδραση του προγράμματος με τον χρήστη και τα δύο αρχεία δεδομένων απεικονίζεται στο παρακάτω σχήμα:



Τα ονόματα των δύο αρχείων δίνονται ως ορίσματα του προγράμματος κατά την εκκίνηση. Το αρχείο της βαθμολογίας μπορεί να υφίσταται από προηγούμενη εκτέλεση του προγράμματος, διαφορετικά πρέπει να δημιουργείται. Το πρόγραμμα πρέπει να ανοίγει τα αρχεία μια μοναδική φορά, και να τα κλείνει πριν τερματιστεί. Οι προσθήκες/αλλαγές πρέπει να σώζονται άμεσα, προτού εκτυπωθεί η επιβεβαίωση προς τον χρήστη. Η πρόσβαση στα αρχεία πρέπει να γίνει χρησιμοποιώντας τις λειτουργίες του συστήματος (όχι της βιβλιοθήκης stdio). Το πρόγραμμα τερματίζεται αν ο χρήστης επιλέξει έξοδο, ή υπάρξει κάποιο σφάλμα (στην οποία περίπτωση πρέπει να εκτυπώνεται και κατάλληλο μήνυμα).

Η αναζήτηση, προσθήκη και διαγραφή των εγγραφών πρέπει να υλοποιηθούν ως ξεχωριστές συναρτήσεις που καλούνται από την main ανάλογα με την λειτουργία που επιλέγει ο χρήστης κάθε φορά. Απαγορεύονται οι καθολικές (global) μεταβλητές και η χρήση goto. Η ζητούμενη λειτουργικότητα πρέπει να υλοποιηθεί χωρίς λίστα/πίνακα εγγραφών στην μνήμη και χωρίς χρήση βοηθητικού αρχείου. Τεκμηριώστε κάθε μεταβλητή και συνάρτηση του προγράμματος. Γενικά, ο κώδικας σας πρέπει να είναι ευανάγνωστος. Κώδικας που «δεν διαβάζεται» θα απορριφθεί, χωρίς εξέταση.

**Για όσους έχουν κέφι/χρόνο (και έχουν υλοποιήσει σωστά τα παραπάνω):** Επεκτείνετε το πρόγραμμα έτσι ώστε να διαχειρίζεται την αναλυτική βαθμολογία πολλών φοιτητών, χρησιμοποιώντας ένα ξεχωριστό αρχείο για κάθε φοιτητή. Ακόμα, το πρόγραμμα πρέπει να δίνει στον χρήστη την δυνατότητα να βλέπει το συνολικό ποσοστό επιτυχίας ενός μαθήματος (#εξετάσεων με βαθμό  $\geq 5$  / #εξετάσεων).

Παράδοση: **Τρίτη 11 Μαρτίου 2014, 22:00**

Οδηγίες παράδοσης: Στην ιστοσελίδα του μαθήματος.