

## Εργασία 1 – Βάση αντικειμένων

Αναπτύξτε ένα πρόγραμμα για την διαχείριση μιας βάσης ψηφιακών αντικειμένων (αντικείμενο = οτιδήποτε μπορεί να αναπαρασταθεί ως μια σειρά από bytes). Η επιθυμητή λειτουργικότητα περιγράφεται παρακάτω.

### Περιγραφή λειτουργίας

Το πρόγραμμα δέχεται ως όρισμα εκκίνησης το όνομα του αρχείου που χρησιμοποιείται για την βάση των αντικειμένων. Αν το αρχείο δεν υπάρχει, δημιουργείται και αρχικοποιείται κατάλληλα. Στην συνέχεια, σε επανάληψη, το πρόγραμμα πρέπει να δέχεται από τον χρήστη και να εκτελεί τις εξής εντολές:

- **i(mport) <name>**: Ο χρήστης δίνει το όνομα του αρχείου που περιέχει το αντικείμενο που θέλει να εισάγει στη βάση. Εκτός από το περιεχόμενο του αντικειμένου, το πρόγραμμα πρέπει να αποθηκεύει στην βάση και το όνομα του αρχείου (χωρίς το μονοπάτι των καταλόγων που πιθανόν προηγείται του ονόματος). Αν το αρχείο που έδωσε ο χρήστης δεν υπάρχει, εκτυπώνεται μήνυμα λάθους. Αν ήδη υπάρχει στην βάση ένα αντικείμενο με το ίδιο όνομα, τότε εκτυπώνεται μήνυμα λάθους.
- **f(ind) <name>**: Ο χρήστης δίνει το όνομα του αντικειμένου που αναζητά. Το πρόγραμμα ψάχνει για όλα τα αντικείμενα στη βάση με ονόματα που εμπεριέχουν (σε οποιοδήποτε σημείο) το όνομα που έδωσε ο χρήστης, και εκτυπώνει τα πλήρη ονόματα τους. Εάν ο χρήστης δώσει ως όνομα το "\*" τότε το πρόγραμμα πρέπει να εκτυπώνει τα ονόματα όλων των αντικειμένων που υπάρχουν στην βάση.
- **e(xport) <src> <dest>**: Ο χρήστης δίνει το όνομα του αντικειμένου προς εξαγωγή και το όνομα του αρχείου στο οποίο θέλει να το αποθηκεύσει. Το πρόγραμμα εντοπίζει το αντικείμενο στην βάση και αντιγράφει τα περιεχόμενα του στο αρχείο που υπέδειξε ο χρήστης. Αν το αντικείμενο δεν υπάρχει στην βάση, εκτυπώνεται μήνυμα λάθους. Αν το αρχείο που υπέδειξε ο χρήστης δεν υπάρχει, δημιουργείται από το πρόγραμμα. Αν αυτό ήδη υπάρχει, εκτυπώνεται μήνυμα λάθους. Σε κάθε περίπτωση, το αντικείμενο παραμένει στην βάση.
- **d(elte) <name>**: Ο χρήστης δίνει το όνομα του αντικειμένου που θέλει να διαγράψει από τη βάση. Το πρόγραμμα εντοπίζει το αντικείμενο και το διαγράφει. Αν αυτό δεν υπάρχει, εκτυπώνεται μήνυμα λάθους.
- **q(uit)**: Εντολή τερματισμού. Το πρόγραμμα τερματίζει, αφού προηγουμένως κλείσει το αρχείο της βάσης.

### Απαιτήσεις υλοποίησης

Η πρόσβαση στα αρχεία πρέπει να γίνει χρησιμοποιώντας τις λειτουργίες του συστήματος (όχι της βιβλιοθήκης stdio που πρέπει να χρησιμοποιούνται αποκλειστικά για τον διάλογο με τον χρήστη). Αν υπάρξει σφάλμα, το πρόγραμμα πρέπει να τερματίζει αφού πρώτα εκτυπώσει κατάλληλο μήνυμα.

Η αναζήτηση πρέπει να γίνεται στο αρχείο της βάσης, και όχι στη μνήμη. Η μεταφορά των περιεχομένων των αντικειμένων από/προς την βάση πρέπει να γίνεται τμηματικά σε κομμάτια (blocks) μέγιστου μεγέθους 512 bytes.

Κατά την εισαγωγή ενός αντικειμένου, το αρχείο της βάσης θα πρέπει να επεκτείνεται κατάλληλα. Αντίστοιχα, κατά την διαγραφή ενός αντικειμένου, το αρχείο της βάσης θα πρέπει να μειώνεται κατάλληλα, τουλάχιστον κατά το μέγεθος του αντικειμένου που διαγράφηκε.

Το πρόγραμμα πρέπει να εισάγει σε επιλεγμένα σημεία στο αρχείο της βάσης επιπλέον πληροφορία “τύπου” με βάση την οποία να μπορεί σε επόμενη εκτέλεση να ελέγχει αν το αρχείο που δόθηκε ως όρισμα από τον χρήστη έχει την σωστή/αναμενόμενη δομή.

Υλοποιήστε τις επιμέρους λειτουργίες αναζήτησης, εισαγωγής, ανάγνωσης και διαγραφής αντικειμένων σε ξεχωριστές συναρτήσεις, που καλούνται μέσα από την main του προγράμματος.

Απαγορεύονται οι καθολικές μεταβλητές και η χρήση goto και gets. Απαγορεύεται η χρήση βοηθητικού αρχείου για την υλοποίηση λειτουργιών της βάσης. Δώστε περιγραφικά ονόματα σε συναρτήσεις και μεταβλητές, σχολιάστε τις συναρτήσεις και όσα τμήματα του κώδικα δεν είναι ξεκάθαρα, και γράψτε ευανάγνωστα με σωστή στοίχιση. Κώδικας που “δεν διαβάζεται”, απορρίπτεται χωρίς εξέταση.

Συζήτηση/επεξήγηση εργασίας: **Πέμπτη 22/2/2018**, στην ώρα του μαθήματος

Προθεσμία παράδοσης εργασίας: **Κυριακή 11/3/2018, 23:59**

Οδηγίες παράδοσης εργασίας: στη σελίδα του μαθήματος.

**Συστάσεις για την ανάπτυξη του κώδικα**

1. Σχεδιάστε την οργάνωση των δεδομένων στο αρχείο της βάσης, έτσι ώστε να υποστηρίζονται με σχετική ευκολία οι διαδικασίες εισαγωγής, αναζήτησης, εξαγωγής και διαγραφής. Χρησιμοποιήστε χαρτί και μολύβι για να σχεδιάσετε την οργάνωση που έχετε στο μυαλό σας. Μην προχωρήσετε σε επόμενο στάδιο (γράψιμο κώδικα) αν δεν βεβαιωθείτε ότι το σχήμα οργάνωσης του αρχείου υποστηρίζει όλες τις ανάγκες του προγράμματος.
2. Αναπτύξτε τις διάφορες διαδικασίες του προγράμματος σας με την εξής σειρά:
  - (a) εισαγωγή
  - (b) αναζήτηση
  - (c) εξαγωγή
  - (d) διαγραφή
3. Μην προσπαθήσετε να υλοποιήσετε πολλές διαδικασίες ταυτόχρονα. Μην προχωράτε στο επόμενο στάδιο της ανάπτυξης, χωρίς να έχετε δοκιμάσει εκτενώς τον κώδικα που έχετε υλοποιήσει μέχρι εκείνο το σημείο ώστε να βεβαιωθείτε για την ορθότητα του. Κάντε δοκιμές με «οριακές» περιπτώσεις, π.χ., εισαγωγή σε κενή βάση, αναζήτηση σε κενή βάση, αναζήτηση αντικειμένου που δεν υπάρχει στην βάση, αναζήτηση αντικειμένου στην αρχή/τέλος του αρχείου, διαγραφή από την αρχή /τέλος της βάσης, κλπ.
4. Εκτός από τις βασικές λειτουργίες για τις οποίες υποχρεωτικά καλείστε να φτιάξετε ξεχωριστές συναρτήσεις, προσπαθήστε να δημιουργήσετε συναρτήσεις για όλα τα τμήματα του κώδικα που επαναλαμβάνονται σε περισσότερα σημεία, καθώς και για επιμέρους λειτουργίες που θέλετε να ελέγξετε αυτόνομα.