

**ΕΡΓΑΣΤΗΡΙΟ 5****Μέρος Α': Struct, union, enum**

1. Κατεβάστε το πρόγραμμα `shape.c` από την ιστοσελίδα του μαθήματος, στον οποίο υπάρχει υλοποιημένη μόνο η συνάρτηση `main()`. Δεν χρειάζεται να αλλάξετε κάτι μέσα στη `main()`. Το πρόγραμμα μοντελοποιεί σχήματα. Ένα σχήμα μπορεί να είναι είτε ορθογώνιο, είτε τρίγωνο, ή κύκλος με διαφορετικά χαρακτηριστικά. Το πρόγραμμα δημιουργεί 3 σχήματα και τυπώνει τα διάφορα χαρακτηριστικά τους στην οθόνη.
 - a. Ορίστε την κατάλληλη δομή που μοντελοποιεί ένα σχήμα (`shape`) και όποιες άλλες δομές κρίνετε αναγκαίες.
 - b. Προσθέστε τα απαραίτητα πεδία στη δομή `shape` έτσι ώστε να αντιπροσωπεύει είτε ορθογώνιο είτε τρίγωνο ή κύκλο.
 - c. Ποιο είναι το μέγεθος ενός αντικειμένου αυτής της δομής; Γιατί;
 - d. Προσθέστε το κατάλληλο κώδικα που μειώνει το μέγεθός της.
 - e. Προσθέστε στον κώδικα μια συνάρτηση `void printShape(shape)`; η οποία τυπώνει τα χαρακτηριστικά ενός (οποιοδήποτε) σχήματος στην οθόνη.
2. Γράψτε ένα πρόγραμμα `sparseMatrix.c` το οποίο μοντελοποιεί ένα αραιό πίνακα δυο διαστάσεων (`sparse matrix`). Αραιός πίνακας θεωρείται ο πίνακας του οποίου τα περισσότερα στοιχεία είναι `=0` και πολύ λίγα `≠0`. Για αυτό το πρόγραμμα θεωρείστε ότι το μέγιστο πλήθος των στοιχείων που είναι `≠0` είναι 101. Το πρόγραμμα να περιέχει τουλάχιστον τρεις συναρτήσεις τα πρότυπα των οποίων είναι τα παρακάτω:

```
int readMatrix(term []);  
void printMatrix(term [], int);  
int search(term [],int, int);
```
3. Κατεβάστε το αρχείο `fun_struct.c` από την ιστοσελίδα του μαθήματος.
 - a. Στο πρόγραμμα αυτό θα ορίσετε και θα διαχειριστείτε ένα σύνθετο τύπο δεδομένων που θα περιέχει πληροφορίες φοιτητών ενός μαθήματος. Σκοπός είναι να υλοποιηθεί ένα αρχείο δεδομένων που να περιέχει πληροφορίες φοιτητών που παρακολουθούν ένα μάθημα. Συγκεκριμένα, πρέπει να ορίσετε ένα νέο σύνθετο τύπο δεδομένων `students`, ο οποίος να περιέχει βασικούς τύπους που να αντιπροσωπεύουν την ταυτότητα ενός φοιτητή, το όνομα του, το επίθετό του, τη βαθμολογία του στις εργασίες (0-10), στην ενδιάμεση εξέταση (0-10), στην τελική εξέταση (0-10) και τον μέσο όρο των βαθμών του (0-10). Να γίνεται έλεγχος για ορθή καταχώρηση των βαθμολογιών εργασίας, ενδιάμεσης εξέτασης και τελικής εξέτασης.
 - b. Να υλοποιηθεί συνάρτηση `void read_students()` μέσω της οποίας ο χρήστης πρέπει να δίνει ως δεδομένα εισόδου το πλήθος των φοιτητών και στη συνέχεια την ταυτότητα, το όνομα, το επίθετο, τη βαθμολογία των εργασιών, ενδιάμεσης και τελικής εξέτασης για κάθε φοιτητή. Οι πληροφορίες αυτές πρέπει



να αποθηκεύονται σ' ένα μονοδιάστατο πίνακα που θα έχει οριστεί στη συνάρτηση `main()`. Στη συνάρτηση `main()` είναι επίσης δηλωμένη η μεταβλητή (`numstud`) που θα περιέχει το πλήθος των φοιτητών.

- c. Να υλοποιηθεί συνάρτηση `void evaluate_final()` η οποία να υπολογίζει και καταχωρεί το μέσο όρο των βαθμών του κάθε φοιτητή.
- d. Να υλοποιηθεί συνάρτηση `void print_students()` η οποία να τυπώνει τα στοιχεία του κάθε φοιτητή όπως φαίνονται στο πιο κάτω παράδειγμα.

Παράδειγμα εκτέλεσης:

```
Dwse to plithos twn foititwn (metaxy 1 kai 5) : 3
Foititis 1
Arithmos tavgotitas: 1012865
Onoma: Antreas
Epitheto: Antreou
1os vathmos foititi: 8
2os vathmos foititi: 9.5
3os vathmos foititi: 8.5
Foititis 2
Arithmos tavgotitas: 10321456
Onoma: Panagiotis
Epitheto: Dimitriou
1os vathmos foititi: 6.5
2os vathmos foititi: 7
3os vathmos foititi: 8
Foititis 3
Arithmos tavgotitas: 909812
Onoma: Giorgos
Epitheto: Xenofontos
1os vathmos foititi: 8.5
2os vathmos foititi: 9
3os vathmos foititi: 12
Edwses lathos vathmo. Xanadwse 3o vathmo foititi: 10
-----
Arithmos tavgotitas 1ou foititi: 1012865
Onoma foititi: Antreas
Epitheto foititi: Antreou
Vathmoi foititi: 8.00 9.50 8.50
Mesos oros mathimatwn foititi: 8.67
-----
Arithmos tavgotitas 2ou foititi: 10321456
Onoma foititi: Panagiotis
Epitheto foititi: Dimitriou
Vathmoi foititi: 6.50 7.00 8.00
Mesos oros mathimatwn foititi: 7.17
-----
Arithmos tavgotitas 3ou foititi: 909812
Onoma foititi: Giorgos
Epitheto foititi: Xenofontos
Vathmoi foititi: 8.50 9.00 10.00
Mesos oros mathimatwn foititi: 9.17
```

**Μέρος Β': Δυναμική δέσμευση μνήμης: malloc(), realloc(), free()**

4. Γράψτε ένα πρόγραμμα, `readShuffleLines.c`, το οποίο διαβάζει γραμμές κειμένου από το `stdin`. Η εισαγωγή να τερματίζεται όταν ο χρήστης να δώσει `end-of-file` (`Ctrl+d` στο `Linux`). Με το πέρας της εισαγωγής δεδομένων, το πρόγραμμα ανακατεύει τυχαία τις γραμμές κειμένου και τις τυπώνει στην οθόνη.
 - Το πρόγραμμα πρέπει να δέχεται όσες γραμμές κειμένου του δίνει ο χρήστης
 - Χρησιμοποιήστε τις συναρτήσεις `srand()` και `rand()` για παραγωγή «τυχαίων αριθμών»
 - Να γίνει σωστή διαχείριση της μνήμης (δέσμευση/malloc, αποδέσμευση/free πόρων)
5. Γράψτε ένα πρόγραμμα, `sortTable.c`, το οποίο χρησιμοποιεί τον αλγόριθμο `insertion sort`, για να ταξινομεί ένα πίνακα με πραγματικούς αριθμούς. Ο αλγόριθμος `insertion sort`, εισάγει ένα στοιχείο στη σωστή θέση σε ένα ήδη ταξινομημένο πίνακα έτσι ώστε ο πίνακας να παραμένει πάντοτε ταξινομημένος. Αρχικά ο πίνακας να δεσμεύεται δυναμικά. Στη συνέχεια το πρόγραμμα να ζητά από το χρήστη να δίνει πραγματικούς αριθμούς. Η εισαγωγή αριθμών να τερματίζεται όταν ο χρήστης να δώσει `end-of-file` (`Ctrl+d` στο `Linux`).
 - Στην αρχή ο πίνακας είναι άδειος και το αρχικό μέγεθος του πίνακα 2
 - Το τελικό μέγεθος του πίνακα είναι αδιευκρίνιστο.
 - Να γίνει σωστή διαχείριση της μνήμης