



## ΕΡΓΑΣΤΗΡΙΟ 8

### Μέρος Α': Macros

1. Στην ιστοσελίδα του προγράμματος κατεβάστε το πρόγραμμα `lab8_example1.c`.
  - Προσθέστε εντολές προς τον επεξεργαστή για κάθε ενότητα εντολών στην `main()`, ώστε να γίνει ξεχωριστά η μεταγλώττιση και η εκτέλεση κάθε ενότητας.
  - Για κάθε συνάρτηση, `int_...()` προσθέστε το αντίστοιχο `macro` στο πρόγραμμα.
  - Συζητήστε με τον υπεύθυνο εργαστηρίου τα αποτελέσματα της εκτέλεσης του προγράμματος.
2. Στην ιστοσελίδα του προγράμματος κατεβάστε τα προγράμματα `lab8_example2.c` και `lab8_example3.c`.
  - Συζητήστε με τον υπεύθυνο εργαστηρίου τον κώδικα και τα αποτελέσματα της εκτέλεσης των προγραμμάτων.
3. Γράψτε ένα πρόγραμμα, `lab8_example4.c`, το οποίο χρησιμοποιεί ένα `macro` για να αλλάζει τις τιμές δυο ακεραίων μεταβλητών. Το ίδιο `macro` να χρησιμοποιηθεί για την ανταλλαγή τιμών δυο μεταβλητών τύπου `float`.

### Μέρος Β': Συνδεδεμένες λίστες

4. Σε αυτή την άσκηση θα πρέπει να υλοποιήσετε πράξεις σχετικά με συνδεδεμένες λίστες.

Δημιουργήστε ένα αρχείο επικεφαλίδων `list.h` στο οποίο θα βάλετε τα πρότυπα συναρτήσεων για μία συνδεδεμένη λίστα πραγματικών αριθμών:

- **`unsigned int length(Node *head)`**  
Επιστρέφει το μέγεθος της συνδεδεμένης λίστας.  
*head*: δείκτης στον πρώτο κόμβο της λίστας.
- **`void print(Node *head)`**  
Εμφανίζει το περιεχόμενο της λίστας στην οθόνη.  
*head*: δείκτης στον πρώτο κόμβο της λίστας.
- **`Node *next(Node *head, Node *of_node)`**  
Επιστρέφει δείκτη στον κόμβο που ακολουθεί ένα άλλο κόμβο στη λίστα.  
*head*: δείκτης στον πρώτο κόμβο της λίστας.  
*of\_node*: δείκτης στον κόμβο του οποίου θέλουμε να βρούμε τον επόμενο κόμβο.



- **Node \*previous(Node \*head, Node \*of\_node)**  
*head*: δείκτης στον πρώτο κόμβο της λίστας.  
*of\_node*: δείκτης στον κόμβο του οποίου θέλουμε να βρούμε τον προηγούμενο κόμβο.
- **Node \*retrieve(Node \*head, unsigned int position)**  
Επιστρέφει ένα δείκτη στον κόμβο της λίστας που βρίσκεται σε μία συγκεκριμένη θέση.  
*head*: δείκτης στον πρώτο κόμβο της λίστας.  
*position*: η θέση του κόμβου μέσα στη λίστα (ξεκινώντας από τη θέση 0).
- **unsigned int locate(Node \*head, unsigned int start\_position, float value)**  
Επιστρέφει τη θέση ενός κόμβου που περιέχει μία συγκεκριμένη τιμή μέσα στη λίστα.  
*head*: δείκτης στον πρώτο κόμβο της λίστας.  
*start\_position*: η θέση από την οποία ξεκινά η έρευνα.  
*value*: η τιμή η οποία ψάχνουμε.
- **void insert(Node \*\*head, Node \*new\_node, unsigned int position)**  
Εισάγει ένα νέο κόμβο στη λίστα.  
*\*head*: δείκτης στον πρώτο κόμβο της λίστας (διπλός δείκτης).  
*new\_node*: δείκτης στο νέο κόμβο.  
*position*: η θέση στην οποία θα μπει ο νέος κόμβος (από 0 έως length(head)).
- **void delete(Node \*\*head, unsigned int position)**  
Διαγράφει ένα κόμβο από τη λίστα.  
*\*head*: δείκτης στον πρώτο κόμβο της λίστας (διπλός δείκτης).  
*position*: η θέση από την οποία θα διαγραφεί ο κόμβος (από 0 έως length(head))

Υλοποιήστε τις συναρτήσεις σε ένα άλλο αρχείο με όνομα `list.c`,

Στο αρχείο `list.c` προσθέστε ένα `driver` αντικειμένου, όπου θα καλούνται οι συναρτήσεις που υλοποιήσατε.

Χρησιμοποιείτε ένα `makefile` για να κάνετε την μεταγλώττιση.