

## Εργαστήριο #9

**Αρχεία:** Για την πρώτη άσκηση, γράψτε τον κώδικα σας στο αρχείο `lab9a.c` που σας δίνουμε. Σε αυτό γίνεται `#include` το αρχείο `lab9.h` που περιέχει τον ορισμό της σταθεράς `DISH_SIZE` και ορίζονται δύο χρήσιμες συναρτήσεις. Για τη δεύτερη άσκηση, γράψτε τον κώδικα σας σε αρχείο με όνομα `lab9b.c`

**Απαιτήσεις και για τις δύο ασκήσεις:** Απαγορεύονται η χρήση `goto`, `gets`, καθολικών ή `static` μεταβλητών, πινάκων μεταβλητού μεγέθους. Μη μεταβάλετε τις προδιαγραφές των συναρτήσεων που καλείστε να υλοποιήσετε. Χρησιμοποιήστε `switch` όπου ενδείκνυται και κατάλληλες δομές επανάληψης. Γράψτε κώδικα που ακολουθεί όλες τις καλές πρακτικές που έχουν αναφερθεί στο μάθημα.

**Διευκρίνιση:** Σε όλα τα μηνύματα, όπου βλέπετε κενό εκτυπώνετε ακριβώς ένα χαρακτήρα ' ' (space).

### Άσκηση 1

Γράψτε ένα πρόγραμμα που προσομοιώνει την προσθήκη αμοιβάδων σε ένα petri dish. Τα στοιχεία μιας αμοιβάδας αναπαρίστανται ως πεδία ενός `struct` και το petri dish αναπαρίσταται ως πίνακας από τέτοια `struct`.

**Ορίστε τους παρακάτω τύπους:**

- Ένα `enum` με όνομα `amoebaFateT` που αναπαριστά το αποτέλεσμα της προσπάθειας εισαγωγής μιας αμοιβάδας στο petri dish, με τιμές `MERGED`, `ERASED`, `ADDED`, `ERROR`.
- Ένα `enum` με όνομα `genusT` που αναπαριστά το γένος μιας αμοιβάδας, με τιμές `UNDA`, `SAPPINIA`, `CHAOS`.
- Ένα `enum` με όνομα `validityT` που αναπαριστά την εγκυρότητα των δεδομένων σε μία θέση του πίνακα αμοιβάδων, με τιμές `VALID`, `INVALID`.
- Ένα `struct` με όνομα `coordsT` που αναπαριστά την ακριβή θέση μιας αμοιβάδας στο petri dish. Περιέχει δύο πεδία τύπου `double`, ένα για κάθε διάσταση, με ονόματα `x` και `y`.
- Ένα `struct` με όνομα `amoebaT` που αναπαριστά μια αμοιβάδα και περιέχει πεδία με ονόματα:
  - `genus` για το γένος της.
  - `pos` για τη θέση της στο petri dish.
  - `size` για το μέγεθος της αμοιβάδας (τύπου `double`).
  - `validity` για την εγκυρότητα των δεδομένων.

**Ροή προγράμματος:**

Το πρόγραμμα κατασκευάζει ένα πίνακα από αμοιβάδες μεγέθους `DISH_SIZE` και τον αρχικοποιεί ώστε το πεδίο `validity` σε όλες τις θέσεις του πίνακα έχει τιμή `INVALID`.

Κατόπιν, σε επανάληψη:

- Καλεί την έτοιμη συνάρτηση** `print_petri_state` για να εκτυπώσει τα περιεχόμενα του πίνακα.
- Εκτυπώνει** το μήνυμα `"More? (y/n)\n"` και **διαβάζει** ένα χαρακτήρα. Αν αυτός είναι 'n', τότε το πρόγραμμα τερματίζει, ενώ αν είναι 'y' προχωρά στο βήμα 3. Διαφορετικά επαναλαμβάνεται το βήμα 2.
- Καλεί μια συνάρτηση** που θα γράψετε εσείς η οποία επιχειρεί να διαβάσει τα στοιχεία μιας αμοιβάδας. Η συνάρτηση:

- Επιστρέφει ακέραιο που υποδεικνύει αποτυχία ή επιτυχία της ανάγνωσης.
  - Παίρνει ως παράμετρο ένα δείκτη προς μια αμοιβάδα στην οποία θα αποθηκεύσει τα στοιχεία που διαβάζονται εντός της συνάρτησης, εφόσον είναι έγκυρα.
  - Εκτυπώνει το μήνυμα "Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS\n" και διαβάζει το γένος της αμοιβάδας ως ακέραιο. Αν δοθεί τιμή που δεν αντιστοιχεί σε έγκυρο γένος, η συνάρτηση επιστρέφει -1. Διαφορετικά, εκτυπώνει το μήνυμα "Enter position:\n" και διαβάζει την τετμημένη και την τεταγμένη της αμοιβάδας με ένα κενό ανάμεσα. Τέλος εκτυπώνει το μήνυμα "Enter size:\n" και διαβάζει το μέγεθος της αμοιβάδας. Αν αυτό δεν είναι θετικό, η συνάρτηση επιστρέφει -1. Εφόσον όλα τα στοιχεία είναι σωστά, επιστρέφει 0.
4. **Ελέγχει** αν η συνάρτηση ανάγνωσης διάβασε επιτυχώς τα στοιχεία μιας αμοιβάδας ή όχι. Αν όχι, εκτυπώνει το μήνυμα "Invalid data.\n" και προχωρά στην επόμενη επανάληψη, διαφορετικά, **καλεί μια συνάρτηση** που θα γράψετε εσείς η οποία διαχειρίζεται την προσπάθεια τοποθέτησης της νέας αμοιβάδας στον πίνακα. Η συνάρτηση:
- Επιστρέφει τιμή τύπου `amoebaFateT`.
  - Παίρνει ως παραμέτρους (α) τον πίνακα αμοιβάδων και (β) την αμοιβάδα προς εισαγωγή
  - Η προσπάθεια τοποθέτησης της νέας αμοιβάδας στον πίνακα γίνεται ως εξής:
    - Εάν βρεθεί μια αμοιβάδα στον πίνακα με απόσταση\* το πολύ 0.5 και ίδιου γένους με την προς-εισαγωγή, τότε οι δύο αμοιβάδες ενώνονται σε μία με μέγεθος ίσο με το άθροισμα των δύο μεγεθών και επιστρέφεται άμεσα `MERGED`. Η τετμημένη και η τεταγμένη της νέας αμοιβάδας παραμένουν ίδιες με αυτής που υπήρχε ήδη στον πίνακα.
    - Εάν η απόσταση είναι το πολύ 0.5 και οι αμοιβάδες είναι διαφορετικού γένους, τότε η μία καταστρέφει την άλλη - δε γίνεται εισαγωγή, παύει να βρίσκεται στον πίνακα η ήδη υπάρχουσα αμοιβάδα κι επιστρέφεται άμεσα `ERASED`. Η θέση του πίνακα που βρισκόταν η αμοιβάδα που καταστράφηκε χαρακτηρίζεται ως άκυρη (`INVALID`).
    - Διαφορετικά, εφόσον υπάρχει διαθέσιμη θέση στον πίνακα, προστίθεται σε αυτόν η νέα αμοιβάδα κι επιστρέφεται άμεσα `ADDED`. Η προσθήκη γίνεται πάντα στην πρώτη διαθέσιμη θέση του πίνακα, ξεκινώντας από την αρχή. (Προσπαθήστε να σκεφτείτε ένα έξυπνο τρόπο να εντοπίσετε αυτή τη θέση, χωρίς να διατρέξετε δεύτερη φορά τον πίνακα.
    - Εάν δεν υπάρχει χώρος στον πίνακα για τη νέα αμοιβάδα, επιστρέφεται `ERROR`.
- \*Για τον υπολογισμό της απόστασης ανάμεσα σε δύο αμοιβάδες, χρησιμοποιήστε μια **συνάρτηση που θα γράψετε εσείς**, που παίρνει ως παραμέτρους δύο μεταβλητές `coordsT` κι επιστρέφει την ευκλείδεια απόσταση ανάμεσα τους.
5. **Εκτυπώνει** ένα μήνυμα ανάλογα με το τι επέστρεψε η συνάρτηση τοποθέτησης νέας αμοιβάδας: Στην περίπτωση `MERGED` εκτυπώνει το μήνυμα "Merged.\n". Στην περίπτωση `ERASED` εκτυπώνει το μήνυμα "Cross-canceled.\n" Στην περίπτωση `ADDED` εκτυπώνει το μήνυμα "Added.\n". Εάν δεν υπήρχε χώρος, εκτυπώνει το μήνυμα "No space.\n".
6. **Εκτυπώνει** το μήνυμα "Total: X\n\n" όπου X το πλήθος αμοιβάδων στον πίνακα. Σκεφτείτε πώς μπορείτε να υπολογίσετε το πλήθος χωρίς να διατρέχετε τον πίνακα επιπλέον φορές.

## Παράδειγμα εκτέλεσης 1

Με κόκκινο η είσοδος του χρήστη. Σε **πορτοκαλί φόντο** επεξηγηματικά σχόλια.

```

More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
1
Enter position:
3.5 2.1
Enter size:
0.6
Added.      Εισαγωγή στη θέση 0
Total: 1

0: ( 3.5,  2.1)  0.6 SAPPINIA

More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
0
Enter position:
4 -1
Enter size:
0.8
Added.      Εισαγωγή στη θέση 1
Total: 2

0: ( 3.5,  2.1)  0.6 SAPPINIA
1: ( 4.0, -1.0)  0.8 UNDA

More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
0
Enter position:
4.1 -0.99
Enter size:
1
Merged.     Απόσταση 3.15 από την 1η, 0.10 από τη 2η. Ενώνεται με τη δεύτερη γιατί είναι ίδιο είδος.
Total: 2

0: ( 3.5,  2.1)  0.6 SAPPINIA
1: ( 4.0, -1.0)  1.8 UNDA

More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
2
Enter position:
3.5 2.2
Enter size:
0.46
Cross-canceled.  Απόσταση 0.10 από τη 1η. Αλληλοκαταστρέφονται.

```

```

Total: 1

1: ( 4.0, -1.0)  1.8 UNDA  Παρατηρήστε πως η θέση 0 είναι κενή/INVALID.

More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
2
Enter position:
12.37 8
Enter size:
0.9
Added.  Εφόσον δεν τίθεται θέμα MERGED/ERASED, εισάγεται στην 1η διαθέσιμη θέση, την 0.
Total: 2

0: (12.4,  8.0)  0.9 CHAOS
1: ( 4.0, -1.0)  1.8 UNDA

More? (y/n)
n

```

## Παράδειγμα εκτέλεσης 2

Με κόκκινο η είσοδος του χρήστη.

```

More? (y/n)
j
More? (y/n)
h
More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
3
Invalid data.  Δεν υπάρχει τιμή ίση με 3 στο enum αποεβαT
More? (y/n)
y
Enter genus: 0=UNDA 1=SAPPINIA 2=CHAOS
0
Enter position:
0 0
Enter size:
0
Invalid data.  Το μέγεθος 0 δε θεωρείται έγκυρο.

More? (y/n)
n

```

## Άσκηση 2

Γράψτε ένα πρόγραμμα που εκτυπώνει με χρήση αναδρομικής συνάρτησης τη δυαδική αναπαράσταση ενός ακεραίου αριθμού ως μια σειρά από ψηφία 0 και 1.

Το πρόγραμμα πρέπει να παίρνει 2 ορίσματα (επιπλέον του ονόματος του).

Εφόσον το δεύτερο όρισμα είναι η συμβολοσειρά `"-bin"` και το τρίτο όρισμα είναι θετικός ακεραίος, το πρόγραμμα καλεί μια συνάρτηση που παίρνει ως παράμετρο ένα απρόσημο ακεραίο και, με αναδρομικό τρόπο, εκτυπώνει μια σειρά από ψηφία 0 και 1 που αποτελούν τη δυαδική του αναπαράσταση. Η συνάρτηση επιστρέφει το πλήθος των ψηφίων 1 που περιέχονται στη δυαδική αναπαράσταση του αριθμού.

Στη συνέχεια, το πρόγραμμα εκτυπώνει την τιμή επιστροφής της συνάρτησης με χαρακτήρα `'\n'` πριν και μετά.

Επιπλέον, το πρόγραμμα πρέπει να ανιχνεύει σε κατάλληλα σημεία εντός της `main` τις παρακάτω καταστάσεις λάθους και σε κάθε περίπτωση να εκτυπώνει το αντίστοιχο μήνυμα και να τερματίζει:

- Λάθος πλήθος ορισμάτων: `"Incorrect number of arguments.\n"`
- Το δεύτερο όρισμα δεν είναι `"-bin"`: `"Invalid option.\n"`
- Το τρίτο όρισμα δεν είναι θετικός ακεραίος (υποθέστε ότι θα δοθεί εγγυημένα ένας ακεραίος): `"Negative input.\n"`

## Παραδείγματα εκτέλεσης

Με κόκκινο η είσοδος του χρήστη στη γραμμή εντολών

|                                                |                                                               |
|------------------------------------------------|---------------------------------------------------------------|
| <pre>./lab9b -bin 12345 11000000111001 6</pre> | <pre>./lab9b -bins 12345 Invalid option.</pre>                |
| <pre>./lab9b -bin -12345 Negative input.</pre> | <pre>./lab9b -bin 123 45 Incorrect number of arguments.</pre> |

## Προαιρετικά

Σκεφτείτε πώς θα ελέγχατε την περίπτωση που ο χρήστης δίνει μετά το `"-bin"` κάτι που δεν είναι καν ακεραίος (π.χ. μπορεί να είναι δεκαδικός αριθμός ή να περιέχει χαρακτήρες που δεν είναι αριθμητικά ψηφία).

## Υποβολή

Υποβάλετε ένα αρχείο zip με όνομα `lab9.zip` που περιέχει μόνο τα αρχεία `lab9a.c` και `lab9b.c`. Έχετε 10 υποβολές στο autolab.