

3η Διάλεξη – εμφάνιση συναρτήσεων, bool παραστάσεις, έλεγχος ροής if/elseif/else

Για την κάθε μία από τις παρακάτω εργασίες:

(α) βάλτε σχόλια στον κώδικα που εξηγούν τα βήματα της επίλυσης και

(β) δημιουργήστε μέσα στην `main` κώδικα που θα επιδεικνύει την καλή λειτουργία των συναρτήσεων.

Σημείωση: Στον τελικό κώδικα μην χρησιμοποιήσετε την `smPrint` μέσα στις ζητούμενες συναρτήσεις, εκτός εάν το ζητά η εκφώνηση ρητά.

Συμβουλές: Διαβάστε προσεκτικά την εκφώνηση. Επιλέξτε περιγραφικά ονόματα μεταβλητών. Χρησιμοποιήστε καλή στοίχιση (formatting) του κώδικα.

ΠΡΟΣΕΞΤΕ ΟΠΩΣΔΗΠΟΤΕ ΤΑ ΠΑΡΑΚΑΤΩ

Ο τρόπος με τον οποίο πρέπει να υποβάλλετε ερωτήσεις περιγράφεται εδώ:

<https://qna.c-programming.allos.gr/doku.php?id=qna:technical:questions>

Ο τρόπος με τον οποίο πρέπει να υποβάλλετε τον κώδικα των εργασιών στο σύστημα υποβολής περιγράφεται εδώ:

<https://qna.c-programming.allos.gr/doku.php?id=qna:lesson:projects:how-to-submit>

Εργασία 3^α – Game of life , μεμονωμένο κελί

Βαθμός δυσκολίας: **1/3**

Όνομα smProject: **smProject-LIFE**

Περιγραφή

Γνωρίστε το Game of life από τη wikipedia καθώς αυτή είναι η πρώτη από μία σειρά εργασιών:

https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Απλουστευτικά πρόκειται για ένα πίνακα (ή ένα πλέγμα) όπου κάθε κελί του μπορεί να βρίσκεται σε μία από δύο καταστάσεις. Είτε να είναι ζωντανό, είτε όχι.

Εφαρμόζοντας τους παρακάτω κανόνες, όλο το πλέγμα μεταβαίνει σε μία νέα κατάσταση, μία νέα γενιά όπως λέγεται. Οι κανόνες είναι:

- Κάθε ζωντανό κελί με 2 ή 3 ζωντανούς γείτονες (στην προηγούμενη κατάσταση) επιβιώνει (στην νέα κατάσταση)
- Κάθε νεκρό (στην προηγούμενη κατάσταση) κελί με 3 ζωντανούς γείτονες ζωντανεύει (στην νέα κατάσταση)
- Όλα τα άλλα ζωντανά κελιά της προηγούμενης κατάστασης πεθαίνουν στην νέα κατάσταση
- Όλα τα άλλα νεκρά κελιά παραμένουν νεκρά.

1	2	3
8		4
7	6	5

Έτσι προκύπτει η νέα κατάσταση του κελιού στην επόμενη γενιά.

Με τον όρο γείτονες εννοούνται τα 8 κελιά που περιστοιχίζουν το εκάστοτε κελί (βλ. σχήμα – η σειρά αρίθμησης είναι τυχαία).

Εσείς καλείστε να γράψετε τη συνάρτηση:

```
function nextGen( isAlive , aliveNeighbors )
```

η οποία για τη δεδομένη κατάσταση ενός κελιού `isAlive` (`true`=ζωντανό/`false`=νεκρό) και το πλήθος των ζωντανών γειτόνων `aliveNeighbors` (ακέραιος αριθμός 0-8) επιστρέφει την κατάσταση του εν λόγω κελιού στην επόμενη «γενιά».

Εργασία 3^β – Fibonacci

Βαθμός δυσκολίας: **1/3**

Περιγραφή

Η ακολουθία Fibonacci (https://en.wikipedia.org/wiki/Fibonacci_sequence) είναι μία γνωστή ακολουθία η οποία «μαθηματικώς» ορίζεται με αναδρομικό τύπο.

Ζητείται να υλοποιήσετε τη συνάρτηση **fibonacci** η οποία υπολογίζει τον N-οστό όρο της ακολουθίας με δύο τρόπους.

(α) με αναδρομικό τρόπο:

```
function fibonacciR(number)
```

(β) με χρήση βρόχου:

```
function fibonacciL(number)
```

Προσοχή!

i. Όταν ζητηθεί όρος μικρότερος του 1, τότε θα πρέπει να επιστρέφεται ο αριθμός -1 ως ένδειξη λάθους.

ii. Επίσης ως δύο πρώτοι όροι είναι οι τιμές 1 και 1.