

5η διάλεξη –πίνακες, ταξινόμηση

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.

Συμβουλές: Διαβάστε προσεκτικά δύο ή τρεις φορές την εκφώνηση. Επιλέξτε περιγραφικά και αυτοεξηγούμενα ονόματα μεταβλητών. Χρησιμοποιήστε καλή στοίχιση. Κατά τη συγγραφή του προγράμματος προσθέστε όσες `smPrint` χρειαστείτε για να βλέπετε τις τιμές των μεταβλητών ώστε να εντοπίζετε ευκολότερα τυχόν λάθη, πριν παραδώσετε όμως, αφαιρέστε τις “βοηθητικές” και κρατήστε μόνο όσες είναι απαραίτητες.

Σημείωση: Στον τελικό κώδικα μην χρησιμοποιήσετε την `smPrint` μέσα στις ζητούμενες συναρτήσεις.

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 5 – Τρίλιζα

Βαθμός δυσκολίας: **2/3**

Περιγραφή

Καλείστε να γράψετε τις ακόλουθες συναρτήσεις.

Η παρακάτω συνάρτηση θα πρέπει να επιστρέφει αληθές ή ψευδές ως προς το εάν έχει τελειώσει το παιχνίδι. Το παιχνίδι έχει τελειώσει όταν δεν υπάρχει άδεια θέση στη σκακιέρα ή όταν υπάρχει τρίλιζα. Δεν εξετάζουμε την «νόμιμη κατάσταση της σκακιέρας», όπως δεν το κάνει και η `tripletExists`.

```
function isFinished(board)
```

Η παρακάτω συνάρτηση θα πρέπει να επιστρέφει αληθές ή ψευδές ως προς το εάν είναι εφικτό το να παιχτεί η θέση `row, col` δηλαδή εάν η θέση είναι κενή και εντός του `board` και το παιχνίδι δεν έχει τελειώσει. Δεν εξετάζουμε την «νόμιμη κατάσταση της σκακιέρας», όπως δεν το κάνει και η `isFinished`.

```
function isPlayablePos(board, row, col)
```

Η παρακάτω συνάρτηση θα πρέπει να μετατρέπει τη θέση σε αριθμό (π.χ. το 0,2 σε 02 (δηλαδή 2) και το 2,1 σε 21). Εάν η θέση που θα δοθεί είναι εκτός `board` τότε να επιστρέφει -1.

```
function rc2int(row, col)
```

Η παρακάτω συνάρτηση θα πρέπει να ελέγχει εάν υπάρχει ελεύθερη θέση στο `board`, στην οποία εάν παίξει ο παίκτης που έχει σειρά, κερδίζει άμεσα το παιχνίδι. Η τιμή που θα επιστρέφεται θα έχει τη μορφή που επιστρέφει η `rc2int` ή θα είναι -1 εάν δεν βρεθεί τέτοια κίνηση, ενώ εάν το `board` δεν είναι σε σωστή κατάσταση ή έχει τελειώσει η παρτίδα θα πρέπει να επιστρέφεται -2.

```
function winningMove(board)
```

ΣΗΜΕΙΩΣΗ 1: Τις συναρτήσεις αυτές πρέπει να τις υλοποιήσετε με αυτή τη σειρά καθώς η κάθε μία (εκτός φυσικά της πρώτης) θα πρέπει να χρησιμοποιεί τουλάχιστον μία από τις προηγούμενες.

ΣΗΜΕΙΩΣΗ 2: Έχετε τις συναρτήσεις από το παράδειγμα της πρακτικής στο εργαστήριο στη διάθεσή μας και πρέπει να τις χρησιμοποιήσετε. Δηλαδή την

```
function tripletExists(board, player)
```

που επιστρέφει αληθές εάν στο `board` έχει κάνει τρίλιζα ο παίκτης `player`, την

```
function playersTurn(board)
```

που επιστρέφει τον παίκτη που είναι σειρά του να παίξει στο `board` με την υπόθεση ότι ο παίκτης X παίζει πρώτος ή επιστρέφει -1 ως ένδειξη λάθος στησίματος στο `board` και την

```
function canPlay(board, row, col, player)
```

που επιστρέφει αληθές ή ψευδές στο κατά πόσο μπορεί ο παίκτης `player` να παίξει στη θέση `row, col`.