

## 5η διάλεξη – break και continue, πίνακες, ταξινόμηση, αναζήτηση

Για την κάθε μία από τις παρακάτω εργασίες:

(α) χρησιμοποιείτε ένα ξεχωριστό project – το κατάλληλο smProject που δίνεται,

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

(γ) δημιουργήστε μέσα στην smMain κώδικα που θα επιδεικνύει την καλή λειτουργία των συναρτήσεων.

Σημείωση: Στον τελικό κώδικα μην χρησιμοποιήσετε τις `printf`, `smGrid`, `smShowGrid`, `smGrid2Image` μέσα στις ζητούμενες συναρτήσεις.

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 `printf` 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 `printf` μέσα στις ζητούμενες συναρτήσεις.

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

**ΠΡΟΣΟΧΗ!** Μην ξεχάσετε να κατεβάσετε και να χρησιμοποιήσετε τα αντίστοιχα smProject για την κάθε άσκηση!

### 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:

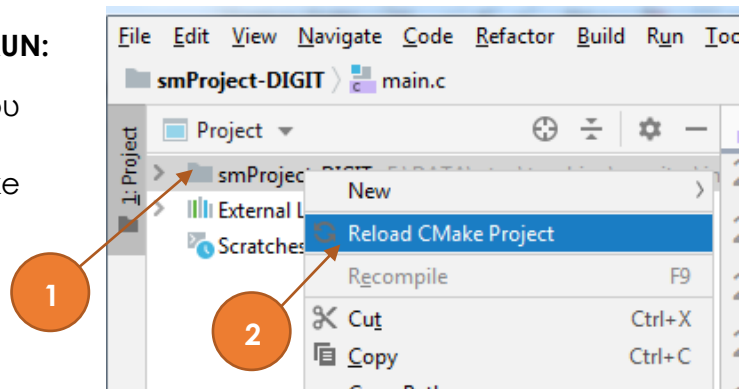
<https://qna.c-programming.allos.gr/doku.php?id=qna:lesson:projects:how-to-submit>

Οι έτοιμες συναρτήσεις της βιβλιοθήκης smLib που διατίθενται για χρήση στα smProjects, περιγράφονται (εκτός από τις διαφάνειες) και στο site ερωταποκρίσεων εδώ:

<https://qna.c-programming.allos.gr/doku.php?id=qna:misc:sm-library>

### ΕΑΝ ΔΕΝ ΕΜΦΑΝΙΖΟΝΤΑΙ ΕΝΕΡΓΑ ΤΑ BUILD / RUN:

1. Κάνω **δεξί κλικ** πάνω στο όνομα του project και εμφανίζεται το μενού
2. Κάνω **απλό κλικ** στο Reload CMake Project, δηλαδή τη 2<sup>η</sup> επιλογή



## Εργασία 5<sup>α</sup> – Τρίλιζα

Βαθμός δυσκολίας: **2/3**

Όνομα smProject: **smProject-TRILIZA**

### 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.

```
bool isFinished(int board[3][3])
```

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.

```
bool isPlayablePos(int board[3][3], int row, int col)
```

Η παρακάτω συνάρτηση θα πρέπει να επιστρέφει αληθές ή ψευδές ως προς το κατά πόσο μπορεί ο παίκτης player να παίξει στη θέση row, col δηλαδή θα πρέπει να επιστρέφει αληθές όταν ταυτόχρονα είναι η σειρά αυτού του παίκτη player να παίξει και μπορεί να παίξει αυτή τη θέση.

```
bool canPlay(int board[3][3], int row, int col, int player)
```

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

```
int rc2int(int row, int 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.

```
int winningMove(int board[3][3])
```

**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

```
bool tripletExists(int board[3][3], int player)
```

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

```
int playersTurn(int board[3][3])
```

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.

## Εργασία 5<sup>B</sup> – Περιστροφή πίνακα

Βαθμός δυσκολίας: 1/3

Όνομα smProject: **smProject-ROTATE**

### Περιγραφή

Καλείστε να γράψετε τη συνάρτηση:

```
bool rotateArray(int A[], int N, int k, bool toRight)
```

η οποία «περιστρέφει» έναν πίνακα  $A$ , μεγέθους  $N$  κατά  $k$  θέσεις προς τα δεξιά (εάν το `toRight` είναι αληθές) ή προς τα αριστερά (όταν το `toRight` είναι ψευδές). Η διαδικασία που πετυχαίνει αυτή την «περιστροφή» περιγράφεται σχηματικά στα παρακάτω βήματα 1,2,3. Δεν είναι ο μόνος τρόπος που γίνεται αυτή η περιστροφή και η επιλογή αυτής της μεθόδου δεν είναι υποχρεωτική.

Εάν η περιστροφή είναι επιτυχής θα πρέπει να επιστρέφει αληθές, εάν όμως δεν είναι (π.χ. εάν το  $N$  ή το  $k$  είναι αρνητικό ή το  $N$  μηδενικό) θα πρέπει να επιστρέφει ψευδές.

**ΣΗΜΕΙΩΣΗ:** Προσέξτε ότι η διαδικασία που φαίνεται είναι εφαρμόσιμη εφόσον το  $k < N$ . Επίσης σκεφθείτε η περιστροφή κατά  $N$  θέσεις που οδηγεί. Με βάση αυτές τις παρατηρήσεις μπορείτε να δείτε πως θα πετυχαίνετε την περιστροφή και κατά  $k > N$ .

