3η διάλεξη – εμβάθυνση συναρτήσεων, bool παραστάσεις, έλεγχος ροής if/elseif/else

Για την κάθε μία από τις παρακάτω εργασίες:

(α) χρησιμοποιείστε ένα ξεχωριστό project – το κατάλληλο smProject που δίνεται για την κάθε εργασία

(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 την `printf` 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.

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:

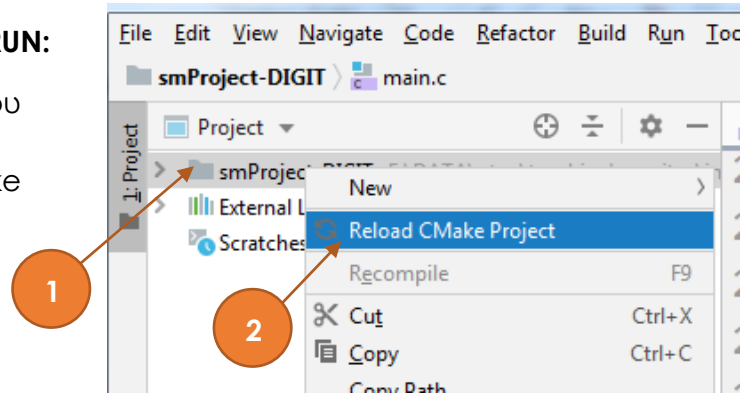
https://qna.c-programming.allos.gr/doku.php?id=qna:technical:questions

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

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 2η επιλογή



Εργασία 3α – Game of life , μεμονωμένο κελί

Βαθμός δυσκολίας: **1/3**

Ονόμα 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 γενεά 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:

```
bool nextGen( bool isAlive , int aliveNeighbors )
```

η οποία για τη δεδομένη κατάσταση ενός κελιού `isAlive` και το πλήθος των ζωντανών γειτόνων `aliveNeighbors` (τα οποία προφανώς δίνονται) επιστρέφει την κατάσταση του εν λόγω κελιού στην επόμενη «γενεά».

Εργασία 3β – Ψηφίο αριθμού

Βαθμός δυσκολίας: **2/3**

Ονόμα smProject: **smProject-DIGIT**

Περιγραφή

Καλείστε να γράψετε τη συνάρτηση `getDigit` που να ελέγχει εάν ένας ακέραιος αριθμός `A` είναι 3ψήφιος (στο δεκαδικό σύστημα) και χρησιμοποιώντας ένα 2ο όρισμα (`digit`) να επιστρέφει το αντίστοιχο ψηφίο του.

```
int getDigit( int A , int digit )
```

Τα ψηφία μετράνε από αριστερά προς τα δεξιά, δηλαδή 1 είναι οι εκατοντάδες και 3 οι μονάδες.

Η συνάρτηση θα πρέπει να επιστρέφει ως ένδειξη λάθους δεδομένων:

- Το -1 εάν ο A δεν είναι 3ψηφιος
- Το -2 εάν το 2ο όρισμα δεν στέκει (<1 ή >3)
- Το -2 εάν παραβιάζονται και οι δύο προϋποθέσεις