

TEM-101 Εισαγωγή στους Η/Υ 2η Εργαστηριακή Άσκηση

Ημερομηνία Παράδοσης: 29 Νοεμβρίου 2010

Σκοπός αυτής της εργαστηριακής άσκησης είναι η συγγραφή ενός προγράμματος το οποίο βοηθάει στην επίλυση του δημοφιλούς παιχνιδιού Sudoku. Αν δεν γνωρίζετε τι είναι το Sudoku μπορείτε να διαβάσετε μια περιγραφή των κανόνων του στη διεύθυνση <http://en.wikipedia.org/wiki/Sudoku>.

Το πρόγραμμα που θα παραδώσετε πρέπει να έχει το όνομα ea2xxxx.c, όπου xxxx είναι ο αριθμός μητρώου σας, και σε σχόλιο στις πρώτες γραμμές του το όνομα και τον αριθμό μητρώου σας. Για παράδειγμα,

```
/*  
 * TEM-101 2h Ergastiriaki Askisi  
 *  
 * Onoma: Michael Plexousakis  
 * A.M. : 9999  
*/
```

Θα πρέπει να το στείλετε με ηλεκτρονικό ταχυδρομείο, από τον προσωπικό σας λογαριασμό, στη διεύθυνση tem101lab@gmail.com, το αργότερο μέχρι τις 21:00, Δευτέρα 29 Νοεμβρίου 2010. Εκπρόθεσμες ασκήσεις δεν θα βαθμολογηθούν. Στα εργαστήρια της εβδομάδας 30/11-3/12 θα πρέπει να είστε σε θέση να απαντήσετε σε ερωτήσεις σχετικά με το πρόγραμμα που παραδώσατε.

Για τις ανάγκες της εργαστηριακής άσκησης θα παραστήσουμε τον πίνακα ενός παιχνιδιού Sudoku με ένα πίνακα ακεραίων διάστασης 9×9 . Κάθε κελί του πίνακα είναι είτε μηδέν, το οποίο δηλώνει ότι η τιμή του αντίστοιχου κελιού δεν έχει ακόμα βρεθεί, είτε ένας αριθμός από το ένα ως το εννέα. Η αρχική κατάσταση του παιχνιδιού δίνεται από τον χρήστη στην είσοδο.

1. Γράψτε τη συνάρτηση `void readBoard(int b[][9])` η οποία διαβάζει 81 ακεραίους και τους αποθηκεύει στον πίνακα `b`. Μεταξύ διαδοχικών ακεραίων υπάρχει κενό ή χαρακτήρας αλλαγής γραμμής. Όπως είπαμε και παραπάνω οι τιμές των ακεραίων κυμαίνονται από 0 ως 9, όπου το 0 σημαίνει πως η αντίστοιχη θέση του πίνακα είναι ακόμα κενή. Μερικοί αρχικοί πίνακες sudoku υπάρχουν στο αρχείο `sudoku.txt`. Μπορείτε να χρησιμοποιήσετε τις παρακάτω εντολές για να διαβάσετε τις τιμές των 81 κελιών:

```
for (i = 0; i < 9; i++)  
    for (j = 0; j < 9; j++)  
        scanf("%d", &b[i][j]);
```

Παρατηρήστε την εντολή `scanf: &b[i][j]` είναι η διεύθυνση του στοιχείου με δείκτη γραμμής `i` και δείκτη στήλης `j`.

2. Γράψτε τη συνάρτηση `void printBoard(int b[][9])` η οποία τυπώνει τον 9×9 πίνακα ενός παιχνιδιού Sudoku που δίνεται ως όρισμα, με τον παρακάτω τρόπο

- Στην πρώτη γραμμή εκτυπώνει τον αύξων αριθμό κάθε στήλης (0 έως και 8). Οι αριθμοί πρέπει να έχουν ίσο κενό μεταξύ τους και όσο κενό χρειάζεται έτσι ώστε ο κάθε αριθμός να βρίσκεται στο κέντρο της στήλης την οποία αντιπροσωπεύει.
- Στη δεύτερη γραμμή εκτυπώνει μια σειρά από χαρακτήρες '='.
- Για κάθε μια από τις επόμενες γραμμές ισχύουν τα ακόλουθα:
 - κάθε κελί πιάνει τρεις θέσεις
 - αν ένα κελί περιέχει μηδέν τότε εμφανίζονται μόνο κενά
 - αν ένα κελί περιέχει αριθμό αυτός εμφανίζεται στο κέντρο του κελιού
 - ανάμεσα σε διαδοχικά κουτάκια υπάρχει ο χαρακτήρας '|'
 - ανάμεσα στις στήλες 2 και 3 και 5 και 6 υπάρχουν δύο χαρακτήρες '|'
 - αριστερά και δεξιά του πίνακα υπάρχει από ένας χαρακτήρας '|'
 - πριν την πρώτη στήλη υπάρχει ο αύξων αριθμός της γραμμής (0 έως και 8) και ένα κενό

- Ανάμεσα σε διαδοχικές γραμμές υπάρχει μια σειρά από χαρακτήρες '-' με την παρακάτω εξαίρεση:
 - ανάμεσα στις γραμμές 2 και 3 και στις γραμμές 5 και 6 υπάρχει μια σειρά από χαρακτήρες '='.
- Στην τελευταία γραμμή υπάρχει μια σειρά από χαρακτήρες '='.
- Οι σειρές από χαρακτήρες '-' ή '=' πρέπει να έχουν τόσο μήκος όσο χρειάζεται για να εμφανιστεί μια γραμμή του πίνακα όπως περιγράφεται παραπάνω

Ακολουθεί ένα παράδειγμα:

	0	1	2	3	4	5	6	7	8					
0				9		8			5		6			
1														
2		8			5					7			2	
3		6				7					3			
4					9			8						
5		1				2					9			
6		3			6					2			4	
7														
8				8		2			9		5			

Βεβαιωθείτε ότι το πρόγραμμά σας εκτυπώνει τον πίνακα στην παραπάνω μορφή προτού προχωρήσετε στο επόμενο βήμα.

3. Η αλληλεπίδραση του προγράμματος με τον χρήστη γίνεται με τον παρακάτω τρόπο:

(α') Εκτυπώνεται η τρέχουσα κατάσταση του πίνακα Sudoku.

(β') Σε νέα γραμμή εκτυπώνεται το μήνυμα

What to you want to do? [1: insert 2: delete 0: quit]

(γ') Διαβάζει την απάντηση του χρήστη η οποία μπορεί να είναι είτε 0 είτε 1 είτε 2. Στην περίπτωση που ο χρήστης δώσει την τιμή 0 το πρόγραμμα θα πρέπει να εκτυπώνει το μήνυμα Thank you. Play again! και να σταματά. Αν η απάντηση του χρήστη είναι 2 τότε ζητάει να σβηστεί το περιεχόμενο ενός κελιού. Το πρόγραμμά σας θα πρέπει να ζητάει τον αριθμό γραμμής και στήλης του στοιχείου το οποίο θέλει να σβήσει, να ελέγχει ότι είναι εντός των ορίων, διαφορετικά να εκτυπώνει ένα κατάλληλο μήνυμα και, τέλος, να αναθέτει την τιμή μηδέν στο αντίστοιχο στοιχείο του πίνακα. Αν η απάντηση του χρήστη είναι 1, τότε αυτός/αυτή ζητάει να εισάγει ένα αριθμό σε κάποιο κελί. Το πρόγραμμά σας σε αυτή την περίπτωση θα πρέπει να διαβάζει τον αριθμό γραμμής και στήλης και την τιμή του στοιχείου και να την αναθέτει στην αντίστοιχη θέση του πίνακα. Φυσικά, θα πρέπει να ελέγχει ότι τόσο ο αριθμός της γραμμής ή στήλης αλλά και το στοιχείο το οποίο εισάγεται είναι μέσα στα επιτρεπτά όρια, διαφορετικά να εκτυπώνει ένα κατάλληλο μήνυμα. Το πρόγραμμα θα πρέπει επίσης να ελέγχει κατά πόσο ο αριθμός που εισάγεται στην συγκεκριμένη θέση παραβιάζει τους κανόνες του Sudoku και να αρνείται την εισαγωγή του αριθμού σε αυτή την περίπτωση. Υπενθυμίζουμε ότι κάθε αριθμός από το 1 έως το 9 πρέπει να εμφανίζεται ακριβώς μια φορά σε κάθε γραμμή, στήλη και 3×3 υποπίνακα. Αν χρήστης δώσει οποιαδήποτε άλλη απάντηση εκτός από 0, 1 ή 2 το πρόγραμμά σας θα πρέπει να εκτυπώνει το μήνυμα Wrong command και να ζητάει εκ νέου εντολή.

Σημείωση: ο ευκολότερος, ίσως, τρόπος να υλοποιήσετε την αλληλεπίδραση με τον χρήστη είναι με χρήση της ανακύκλωσης `while`:

```
int play = 1;
...
while (play) {
    ...
}
```

Βεβαιωθείτε ότι το πρόγραμμά σας μεταγλωττίζεται χωρίς σφάλματα και λειτουργεί σωστά προτού το παραδώσετε. Σε περίπτωση που δεν υλοποιήσετε κάποιες από τις λειτουργίες που ζητάει η άσκηση να το αναφέρετε ρητά σε σχόλιο στην αρχή του προγράμματος. Δείξτε τη δικιά σας δουλειά! Προγράμματα που είναι προϊόντα αντιγραφής θα μηδενιστούν.