



## Θ.Ε. ΠΛΣ50 (2007-2008) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε2

### Στόχος

Στην εργασία αυτή θα ασχοληθούμε με βρόγχους, ελέγχους, δισδιάστατους πίνακες, συναρτήσεις, μορφοποιημένη είσοδο / έξοδο στην οθόνη, είσοδο / έξοδοι σε αρχεία.

### Εισαγωγικά:

Το Sudoku είναι ένα παιχνίδι λογικής που έγινε ιδιαίτερα δημοφιλές μέσα από τις εφημερίδες και τα περιοδικά. Σκοπός του παιχνιδιού είναι να τοποθετηθούν οι αριθμοί από το 1 μέχρι το 9 σε ένα μισο-συμπληρωμένο πίνακα 9 x 9 (τον οποίο θα αποκαλούμε στη συνέχεια Sudoku). Το Sudoku αποτελείται από μικρότερους πίνακες 3 x 3 (υποπίνακες). Όλα τα κελιά πρέπει να συμπληρωθούν με αριθμούς από το 1 μέχρι το 9. Όταν συμπληρώνονται αριθμοί πρέπει να ικανοποιούνται οι ακόλουθες συνθήκες:

- Ένας αριθμός μπορεί να υπάρχει μία μόνο φορά σε κάθε στήλη
- Ένας αριθμός μπορεί να υπάρχει μία μόνο φορά σε κάθε γραμμή
- Ένας αριθμός μπορεί να υπάρχει μία μόνο φορά σε κάθε υποπίνακα 3x3

Ακολουθεί ένα παράδειγμα Sudoku:

*-----*											
(2)	.	(1)		(9)	.	.		.	.	.	
(6)	.	.		.	(5)	(8)		.	(4)	(1)	
.	.	.		.	.	(3)		.	(5)	.	
-----+				-----+							
.	(1)	.		.	.	.		.	(7)	.	
.	(2)	(4)		(3)	.	(5)		(1)	(8)	.	
.	(7)	.		.	.	.		.	(2)	.	
-----+				-----+							
.	(6)	.		(5)	.	.		.	.	.	
(7)	(9)	.		(2)	(4)	.		.	.	(5)	
.	.	.		.	.	(7)		(4)	.	(2)	
*-----*											

Οι τελείες αντιπροσωπεύουν τα κενά κελιά προς συμπλήρωση. Οι αριθμοί εντός των παρενθέσεων τους αρχικά δοσμένους αριθμούς. Ακολουθεί το Sudoku λυμένο. Οι αριθμοί εκτός παρενθέσεων είναι οι αριθμοί που συμπλήρωσε ο παίκτης.

*-----*										
(2)	5	(1)		(9)	6	4		7	3	8
(6)	3	9		7	(5)	(8)		2	(4)	(1)
8	4	7		1	2	(3)		6	(5)	9
-----+-----+										
3	(1)	6		8	9	2		5	(7)	4
9	(2)	(4)		(3)	7	(5)		(1)	(8)	6
5	(7)	8		4	1	6		9	(2)	3
-----+-----+										
4	(6)	2		(5)	8	9		3	1	7
(7)	(9)	3		(2)	(4)	1		8	6	(5)
1	8	5		6	3	(7)		(4)	9	(2)
*-----*										

Σημειώστε ότι έγκυρο (σωστά κατασκευασμένο) θεωρείται ένα Sudoku όταν έχει μοναδική λύση.



Sudoku μπορούν να οριστούν και για άλλες διαστάσεις, π.χ. με μέγεθος 4x4 (μέγεθος υποπινάκων 2x2 και αριθμούς 1-4), με μέγεθος 16x16 (μέγεθος υποπινάκων 4x4 και αριθμούς 1-9 και A-F), κ.ο.κ.

Ο κώδικας που θα φτιάξετε θα πρέπει να περιλαμβάνει τον ορισμό της σταθεράς size, που θα εκφράζει τη διάσταση του Sudoku π.χ.

```
#define size 9
```

για Sudoku 9x9 (με μέγεθος υποπινάκων 3x3 και αριθμούς 1-9). Ο κώδικάς σας θα πρέπει να λειτουργεί για τιμή του size ίση με 4 ή 9. Φυσικά, καθώς το size θα αποτελεί σταθερά, το εκτελέσιμο που θα προκύψει θα λειτουργεί για μία μόνο τιμή του size (αυτή που ορίστηκε στον κώδικα πριν τη μεταγλώττιση).

Ένα άλυτο ή ένα μισο-λυμένο Sudoku θα πρέπει να μπορεί να αποθηκεύεται σε ή να διαβάζεται από αρχείο κειμένου με την ακόλουθη μορφοποίηση. Θα περιλαμβάνονται τόσες γραμμές όσες και οι γραμμές του Sudoku και τόσοι χαρακτήρες σε κάθε γραμμή όσες και οι στήλες του Sudoku. Οι χαρακτήρες θα είναι είτε θετικοί αριθμοί (οι δοσμένοι αριθμοί του Sudoku), είτε μηδενικά (τα κενά κελιά προς συμπλήρωση του Sudoku), είτε μικρά λατινικά γράμματα (που αντιστοιχούν σε αριθμούς που συμπλήρωσε ο χρήστης, a=1, b=2, κοκ). Π.χ. το αρχείο για το παραπάνω άλυτο Sudoku θα είχε το ακόλουθο περιεχόμενο.

```
201900000
600058041
000003050
010000070
024305180
070000020
060500000
790240005
000007402
```

ενώ μια συμπληρωμένη ως προς τα κενά της 1<sup>ης</sup> γραμμής εκδοχή του παραπάνω Sudoku θα είχε το ακόλουθο περιεχόμενο

```
2e19fdgch
600058041
000003050
010000070
024305180
070000020
060500000
790240005
000007402
```

Πρέπει να ορίσετε κατάλληλα τον τύπο του sudoku (π.χ. ως διδιάστατο πίνακα αριθμών ή διδιάστατο πίνακα χαρακτήρων), ώστε να εκτελούνται με ευκολία όλες οι λειτουργίες του προγράμματος. Υπόδειξη: σκεφτείτε κάποιον απλό τρόπο να ξεχωρίζουν οι δοσμένοι αριθμοί, οι αριθμοί που συμπλήρωσε ο παίκτης και τα κενά κελιά. Π.χ. ορίστε τον τύπο του sudoku ως διδιάστατο πίνακα ακεραίων αριθμών (short, ή int), όπου τιμή ίση με 0 σημαίνει κενό κελί, τιμή αρνητική σημαίνει δοσμένος αριθμός ίσος με την απόλυτη τιμή της αρνητικής τιμής και θετικός αριθμός σημαίνει αριθμός που συμπλήρωσε ο χρήστης.



## Θέμα 1: Εμφάνιση Sudoku

Κατασκευάστε μια συνάρτηση

```
void display_puzzle(sudoku_type sudoku);
```

που εμφανίζει ένα sudoku στην οθόνη, με τη μορφοποίηση που ακολουθεί και το παραπάνω παράδειγμα.

## Θέμα 2: Ανάγνωση / Αποθήκευση Sudoku

Σας δίνεται έτοιμος ο κώδικας μιας συνάρτησης που διαβάζει ένα άλυτο ή ένα μισο-λυμένο sudoku από αρχείο.

```
/* read a sudoku from a file */
/* and store it at sudoku data structure, where 0 = empty cell, */
/* negative = given number, positive = player's number */
int read_puzzle(sudoku_type sudoku, FILE *in) {
    int i,j;
    int c;
    for (i=0;i<size;i++) {
        for (j=0;j<size;j++){
            c = fgetc(in);
            if (c == EOF)
                return 0;
            else if (c == '0')
                /* if c is zero character, store 0 =empty cell */
                sudoku[i][j] = 0;
            else if (c<= '0'+size && c>= '1')
                /* if c is a character representing */
                /* a given number, transform */
                /* character to negative number */
                sudoku[i][j] = '0'-c;
            else if (c <= 'a'+size-1 && c >= 'a')
                /* if c is a character representing */
                /* a player's number, transform */
                /* character to positive number */
                sudoku[i][j] = c-'a'+1;
            else
                return 0;
        }
        fgetc(in);
    }
    return 1;
}
```

Κατασκευάστε και μια συνάρτηση

```
int write_puzzle(sudoku_type sudoku, FILE *out);
```

που αποθηκεύει ένα άλυτο ή ένα μισο-λυμένο sudoku σε αρχείο. Η συνάρτηση αυτή να επιστρέφει μηδενική τιμή σε περίπτωση σφάλματος και μη μηδενική τιμή διαφορετικά.



## Θέμα 3: Συμπλήρωση Sudoku

Κατασκευάστε μια συνάρτηση

```
int play(sudoku_type sudoku);
```

που επαναλαμβανόμενα δέχεται τιμές από το χρήστη για τα κενά κελιά του sudoku, μέχρι να επιλυθεί το sudoku. Πιο συγκεκριμένα, η αλληλεπίδραση με το χρήστη θα πρέπει να είναι η ακόλουθη:

- Θα ζητείται από το χρήστη να δώσει τιμή σε κάποιο ελεύθερο τετράγωνο με τη μορφή π.χ. 4,5=9, δηλ. στο κελί που βρίσκεται στη 4<sup>η</sup> γραμμή και 5<sup>η</sup> στήλη βάλει τον αριθμό 9. Αν ο χρήστης δώσει εντολή σε άλλη μορφή ή μη έγκυρη τιμή, θα πρέπει να εμφανίζεται ένα κατάλληλο μήνυμα και να ζητείται να δώσει νέα εντολή εισαγωγής.
- Στη συνέχεια, γίνεται έλεγχος αν η τιμή που έδωσε ο χρήστης μπορεί να εισαχθεί στο συγκεκριμένο κελί βάσει των κανόνων του sudoku (υπόδειξη: μπορείτε να φτιάξετε μια ξεχωριστή συνάρτηση για να κάνει τον έλεγχο αυτό, την οποία θα καλείτε από την play). Αν παραβιάζεται κάποιος κανόνας ή αν το κελί αυτό περιέχει ήδη κάποιο άλλο αριθμό ή κάποιο δοσμένο αριθμό, θα πρέπει να εμφανίζεται ένα κατάλληλο μήνυμα και να ζητείται να δώσει νέα εντολή εισαγωγής.
- Στη συνέχεια, η συνάρτηση θα ελέγχει αν με την εισαγωγή του αριθμού που έγινε συμπληρώθηκε το sudoku (υπόδειξη: μπορείτε να φτιάξετε μια ξεχωριστή συνάρτηση για να κάνει τον έλεγχο αυτό, την οποία θα καλείτε από την play – η υπόδειξη αυτή όπως και η προηγούμενη έχουν ως στόχο να τονίσουν ότι μια συστηματική αντιμετώπιση της λύσης ενός προβλήματος περιλαμβάνει τη διάσπαση του προβλήματος σε μικρότερα ανεξάρτητα προβλήματα που κατά κανόνα κωδικοποιούμε σε ξεχωριστές συναρτήσεις). Τότε, πρέπει να εμφανίζεται μήνυμα που να λέει ότι ολοκληρώθηκε το sudoku επιτυχώς και η συνάρτηση να επιστρέφει 1 (= επιτυχία). Αλλιώς, πρέπει να εμφανίζεται το sudoku, όπως έχει διαμορφωθεί μετά την εισαγωγή του νέου αριθμού, και να ζητείται από το χρήστη μια νέα εντολή εισαγωγής.
- Αν ο χρήστης δώσει ως εντολή εισαγωγής 0,0=0 σημαίνει ότι ο χρήστης θέλει να σταματήσει να παίζει. Τότε η συνάρτηση θα επιστρέφει 0 (= εγκατάλειψη).
- Αν ο χρήστης δώσει την εντολή π.χ. 4,5=0 αυτό σημαίνει ότι ο χρήστης θέλει να «καθαρίσει» την τιμή αυτού του κελιού. Φυσικά δεν επιτρέπεται να αλλοιωθούν οι δοσμένοι αριθμοί (σε μια τέτοια απόπειρα θα πρέπει να εμφανίζεται το κατάλληλο μήνυμα).

Υπόδειξη: στη συνέχεια σας δίνεται συνοπτικός κώδικας που διαβάζει γραμμές με εντολές του τύπου 4,5=9 και ελέγχει μη έγκυρες μορφές.

```
if (scanf("%d,%d=%d",&i,&j,&val) != 3) {  
    while (getchar() != '\n');  
    printf("wrong format of command\n");  
} else
```

Αν η scanf δεν επιστρέψει 3, σημαίνει ότι δε μπόρεσε να διαβάσει με επιτυχία 3 ακεραίους, όπου ο πρώτος χωρίζεται από τον δεύτερο με κόμμα και ο τρίτος χωρίζεται από τον δεύτερο με ίσον. Σε αυτή την περίπτωση, διαβάζονται και αγνοούνται οι χαρακτήρες μέχρι την επόμενη αλλαγή γραμμής (μέχρι του σημείου που είχε πατήσει ENTER ο χρήστης) και ενημερώνεται ο χρήστης για το λάθος. Αν η scanf επιστρέψει 3, η ανάγνωση ήταν επιτυχής και θα εκτελεστεί η εντολή που ακολουθεί το else. Μπορείτε να χρησιμοποιήσετε τον κώδικα αυτόν, ή να φτιάξετε αντίστοιχο δικό σας.



## Θέμα 4: Κατασκευή ολοκληρωμένου προγράμματος

Κατασκευάστε ένα συνολικό πρόγραμμα που θα αξιοποιεί τις παραπάνω συναρτήσεις ως εξής:

- Θα ζητείται από το χρήστη ένα όνομα αρχείου που περιέχει ένα έγκυρο sudoku και θα καλείται η `read_puzzle()` για αυτό το αρχείο
- Θα εμφανίζεται το sudoku που διάβασε η `read_puzzle()` στην οθόνη και θα καλείται η `play()`.
- Είτε η `play()` τερματιστεί επειδή ο χρήστης έλυσε το sudoku, είτε επειδή ο χρήστης εγκατέλειψε, θα ζητείται από το χρήστη ένα όνομα αρχείου και θα καλείται η `write_puzzle()` για αυτό το αρχείο.

### Παρατηρήσεις:

Ο κώδικάς σας πρέπει να συμβαδίζει με το πρότυπο ANSI C, να περιλαμβάνει εύστοχα και περιεκτικά σχόλια, να έχει καλή στοίχιση και το όνομα κάθε μεταβλητής, σταθεράς, ή συνάρτησης να είναι ενδεικτικό του ρόλου της.

Σας δίνεται και ένα σύνολο από sudoku (gamefiles.zip), ώστε να δοκιμάσετε τον κώδικά σας.

Κριτήρια αξιολόγησης:

Θέμα 1: Εμφάνιση Sudoku	30
Θέμα 2: Ανάγνωση / Αποθήκευση Sudoku	15
Θέμα 3: Συμπλήρωση Sudoku	35
Θέμα 4: Κατασκευή ολοκληρωμένου προγράμματος	20
Γενική εικόνα (ευανάγνωστος κώδικας, σχολιασμός, κλπ.)	10
<b>ΣΥΝΟΛΟ</b>	<b>110</b>

Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας (με μέγιστη τιμή το 10).

## Τρόπος – Ημερομηνία Παράδοσης

Η εργασία σας θα πρέπει να έχει φτάσει στον Καθηγητή-Σύμβουλό σας μέχρι **την Κυριακή 11/11/2007**.

Περιμένουμε όλες οι εργασίες να αποσταλούν μέσω Email και να είναι γραμμένες σε επεξεργαστή κειμένου MSWord (διευκρινήσεις, σχόλια και παραδοχές που θεωρείτε απαραίτητα να συμπεριλάβετε στην εργασία σας). Τα τμήματα κώδικα και τα τμήματα δεδομένων θα βρίσκονται σε ξεχωριστά αρχεία και θα υπάρχουν αναφορές σ' αυτά από το κείμενο της εργασίας. Στον Καθηγητή-Σύμβουλό σας, σε κάθε περίπτωση, στέλνετε ΕΝΑ μόνο αρχείο (συμπιεσμένο).

Δεν θα δοθεί παράταση στην παράδοση της εργασίας για κανένα λόγο. Την Τρίτη 13/11/2007 ώρα 15:00, θα δημοσιευθεί ενδεικτική επίλυση της εργασίας στο διαδίκτυο (portal).

**Καλή Επιτυχία!!!**