



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ 232 – Προγραμματιστικές Τεχνικές και Εργαλεία

Άσκηση 3 – Αυτόματη Επίλυση του Παιχνιδιού Λατινικά Τετράγωνα

Διδάσκων: Ανδρέας Αριστείδου

Υπεύθυνοι Εργαστηρίων: Παύλος Αντωνίου & Πύρρος Μπράτσкас

Ημερομηνία Ανάθεσης: 4 Νοεμβρίου 2024

Ημερομηνία Παράδοσης: 25 Νοεμβρίου 2024 ώρα 13:00

(ο κώδικας να υποβληθεί σε ένα zip μέσω του Moodle)

I. Στόχοι

Στην εργασία αυτή θα ασχοληθείτε με **δυναμικές δομές δεδομένων**. Συγκεκριμένα, καλείστε να υλοποιήσετε ένα αλγόριθμο οπισθοδρόμησης (backtracking algorithm) με χρήστη στοίβας για να **επιλύσετε αυτόματα** το παιχνίδι λατινικών τετραγώνων που είδατε στην πρώτη άσκηση. Η υλοποίηση θα γίνει σε ομάδες των δυο (2) ατόμων. Για την υλοποίηση της άσκησης θα χρειαστεί να ακολουθήσετε:

1. Διάσπαση του προγράμματος σε πολλαπλά αρχεία .c και .h με χρήση generic **makefile** μαζί με τα σχετικά αρχεία κεφαλίδας. Για παράδειγμα μπορείτε να έχετε αρχεία stack.h/.c που να σχετίζονται με τη χρήση στοίβας, τα αρχεία file.h/.c που να σχετίζονται με το αρχείο εισόδου, κτλ.
2. Το πρόγραμμα πρέπει να μεταγλωττίζεται μέσω της γραμμής εντολής με το Makefile.
3. **Κάθε αντικείμενο (module)** πρέπει να συμπεριλαμβάνει τον **σχετικό οδηγό χρήσης (driver function)**, δείτε διάλεξη 12) – main function μέσα σε #ifdef / #endif.
4. Το τελικό λογισμικό σας θα πρέπει να ελεγχθεί με valgrind για **διαρροή μνήμης**.
5. **Σχόλια** και οδηγό σχολίων με χρήση του **doxygen** αλλά και διάγραμμα εξαρτήσεων αντικειμένων με χρήση του **graphviz** διαθέσιμο στις μηχανές των εργαστηρίων.

Θα ξεκινήσουμε με την περιγραφή του αλγόριθμου και στη συνέχεια θα δώσουμε τα ζητούμενα.

II. Περιγραφή παιχνιδιού λατινικών τετραγώνων

Το παιχνίδι λατινικών τετραγώνων περιεγράφηκε στην άσκηση 1. Ένα λατινικό τετράγωνο τάξης n είναι ένας πίνακας $n \times n$ με ακριβώς n διαφορετικούς αριθμούς από 1 έως n όπου κάθε αριθμός εμφανίζεται μία φορά σε κάθε γραμμή και μία φορά σε κάθε στήλη. Ένα παράδειγμα λατινικού τετραγώνου τάξης 4 φαίνεται στο σχήμα 1. Στα αριστερά δίνεται ένα ασυμπλήρωτο λατινικό τετράγωνο και στα δεξιά η λύση του.

	2		4
2	3		
		1	2
			3

1	2	3	4
2	3	4	1
3	4	1	2
4	1	2	3

Σχήμα 1: Αρχική διάταξη (αριστερά) και τελική διάταξη (δεξιά) παιχνιδιού.

III. Ζητούμενα Άσκησης

Ζητούμενο αυτής της άσκησης είναι η κατασκευή ενός προγράμματος στη γλώσσα προγραμματισμού C που θα λύνει λατινικά τετράγωνα **για τα διάφορες τιμές του n όπως αυτό προσδιορίζεται στο αρχείο εισόδου**, χρησιμοποιώντας τον δοθέντα αλγόριθμο οπισθοδρόμησης ο οποίος θα περιγραφεί πιο κάτω.

Βήματα αλγόριθμου οπισθοδρόμησης με χρήση στοίβας

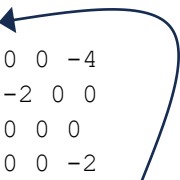
1. Διαβάστε το αρχείο δεδομένων και αποθηκεύσετε σε **δυναμικό** πίνακα δύο διαστάσεων (απαγορεύεται η χρήση στατικών πινάκων). Ο πίνακας αυτός αναπαριστά την αρχική κατάσταση του παιχνιδιού.
2. Αναζητείστε την πρώτη κενή θέση στον πίνακα αρχικής κατάστασης όπου θα μπει ένας αριθμός από 1 έως n
3. Επαναλάβετε τα πιο κάτω:
 - a. Δοκιμάστε να βρείτε ένα αριθμό που αν τοποθετηθεί στην επιλεγθείσα θέση του πίνακα δεν παραβιάζει κανένα κανόνα (ένας αριθμός εμφανίζεται μία φορά σε κάθε γραμμή και μία φορά σε κάθε στήλη).
 - b. Αν βρεθεί αριθμός (που δεν παραβιάζει κανένα κανόνα), τοποθετήστε τον στον πίνακα. Μετά την τοποθέτηση δημιουργείται ο νέος πίνακας τρέχουσας κατάστασης. Εισάγετε στη στοίβα (push) τον νέο πίνακα τρέχουσας κατάστασης και τη θέση (γραμμή και στήλη) που τοποθετήσατε τον αριθμό. Στη συνέχεια, αναζητήστε την επόμενη κενή θέση στον πίνακα τρέχουσας κατάστασης.
 - c. Αν δεν βρεθεί αριθμός (παραβιάζεται κάποιος κανόνας για κάθε αριθμό), διαγράφουμε τον πίνακα τρέχουσας κατάστασης του παιχνιδιού, και εξάγουμε την προηγούμενη κατάσταση που είναι αποθηκευμένη στη στοίβα (pop) και γίνεται αυτή η νέα τρέχουσα κατάσταση. Εξάγουμε επίσης από τη στοίβα τη θέση (γραμμή και στήλη) που μπήκε ο τελευταίος αριθμός έτσι ώστε να δοκιμάσουμε τον επόμενο αριθμό που μπορεί να μπει στη θέση αυτή ή σε κάποια επόμενη διαθέσιμη θέση του πίνακα.

Μέχρι να επιλυθεί το παιχνίδι ή να φτάσουμε σε σημείο που δεν έχει λύση οπότε το πρόγραμμα τερματίζεται.

Παράδειγμα

Όπως στην άσκηση 1, το αρχείο εισόδου θα περιέχει στην πρώτη γραμμή την τάξη του παιχνιδιού n και στις επόμενες γραμμές τα στοιχεία του λατινικού τετραγώνου. Έστω το αρχείο εισόδου lsq2.txt περιέχει:

```
4
0 0 0 -4
0 -2 0 0
0 0 0 0
0 0 0 -2
```



Η πρώτη γραμμή περιέχει την τάξη (αριθμός γραμμών και στηλών) του παιχνιδιού λατινικά τετράγωνα. Αρχικά διαβάζεται το αρχείο και δημιουργείται **δυναμικός** πίνακας 2 διαστάσεων 4×4 ο οποίος περιέχει την αρχική κατάσταση του παιχνιδιού όπως φαίνεται πιο κάτω:

			4
	2		
			2

Σχήμα 2: Πίνακας με την αρχική κατάσταση του παιχνιδιού όπως διαβάζεται από το αρχείο του παραδεί

Στη συνέχεια αναγνωρίζεται ότι η πρώτη κενή θέση είναι η $(0,0)$. Δοκιμάζονται οι αριθμοί από 1 έως $n(=4)$ στη θέση $(0,0)$. Επιλέγεται ο αριθμός 1 που δεν παραβιάζει κάποιο κανόνα οπότε ο νέος πίνακας κατάστασης και η θέση $(0,0)$ στην οποία τοποθετήθηκε ο αριθμός μπαίνουν στη στοίβα που φαίνεται πιο κάτω. Αναζητείται και η νέα κενή θέση που είναι η $(0,1)$.

1			4
	2		
			2

0, 0

Σχήμα 3: Στοιίβα μετά την εισαγωγή της κατάστασης παιχνιδιού και της θέσης που εισήχθηκε αριθμός.

Στη συνέχεια, δοκιμάζονται οι αριθμοί από 1 έως $n(=4)$ στη θέση (0,1). Επιλέγεται ο αριθμός 3 που δεν παραβιάζει κάποιο κανόνα όποτε ο νέος πίνακας κατάστασης και η θέση (0,1) στην οποία τοποθετήθηκε αριθμός μπαίνουν στη στοιίβα που φαίνεται πιο κάτω. Αναζητείται και η νέα κενή θέση που είναι η (0,2).

1	3		4
	2		
			2

0, 1

1			4
	2		
			2

0, 0

Σχήμα 4: Στοιίβα μετά την εισαγωγή της κατάστασης παιχνιδιού και της θέσης που εισήχθηκε αριθμός.

Στα επόμενα βήματα εισάγονται διαδοχικά οι αριθμοί: 2 στη θέση (0,2), 3 στη θέση (1,0) και 1 στη θέση (1,3) και η στοιίβα γίνεται:

1	3	2	4
3	2	1	
			2

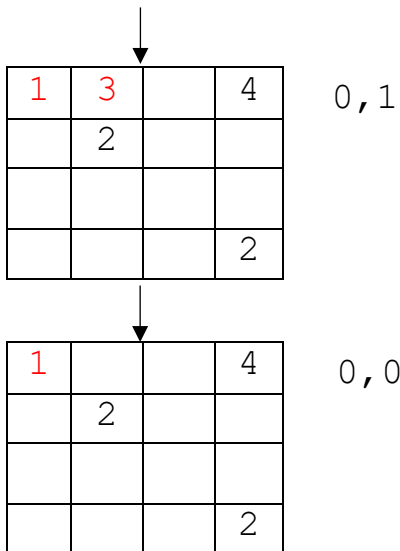
1, 2

1	3	2	4
3	2		
			2

1, 0

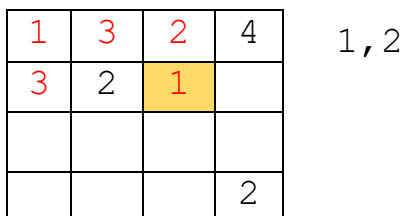
1	3	2	4
	2		
			2

0, 2



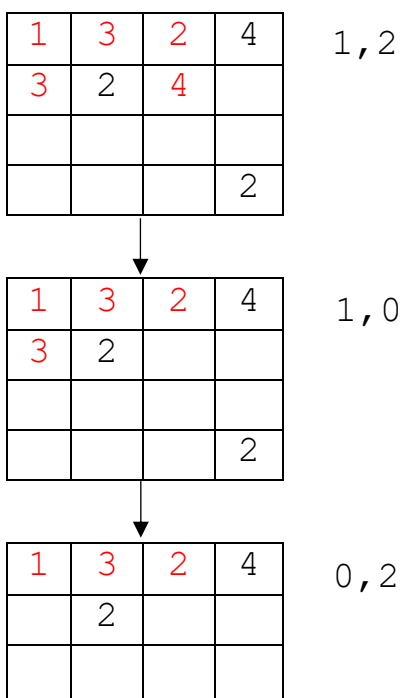
Σχήμα 5: Στοιβά μετά την εισαγωγή 5 καταστάσεων παιχνιδιού.

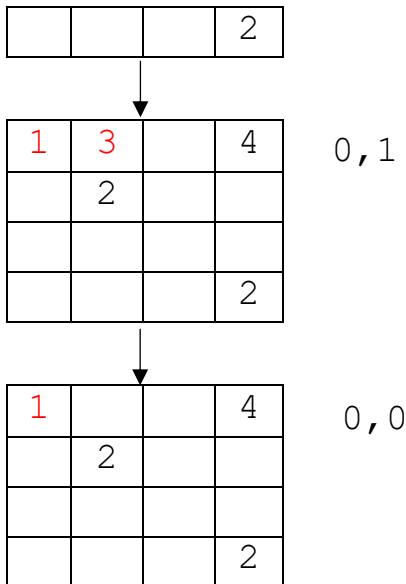
Στο επόμενο βήμα υπάρχει πρόβλημα αφού στη θέση (1,3) δεν υπάρχει αριθμός που να μπορεί να μπει χωρίς να παραβιάζεται κανένας κανόνας. Οπότε εξάγεται (pop) η τελευταία κατάσταση που υπάρχει στη στοιβά όπου είχε τοποθετηθεί το 1 στη θέση (1,2).



Σχήμα 6: Τα δεδομένα που εξάγονται από τη στοιβά.

Στη συνέχεια θα δοκιμαστεί αν μπορεί να μπει άλλος αριθμός στη θέση (1,2) που φαίνεται με κίτρινο σκιασμένο χρώμα στο σχήμα 6. Συνεχίζοντας μετά το 1 (που απορρίφθηκε), το 2 και το 3 δεν μπορούν να μπουν (λόγω παραβίασης κανόνων) αλλά μπορεί να μπει το 4 και έχουμε τη νέα κατάσταση που καταχωρείται στη στοιβά. Η νέα κατάσταση της στοιβάς φαίνεται πιο κάτω:





Σχήμα 7: Η νέα κατάσταση της στοίβας μετά την επιτυχή καταχώρηση του 4 στη θέση (1,2).

Υλοποίηση

Το πρόγραμμα θα δέχεται ως παράμετρο (command line argument) το όνομα του αρχείου εισόδου που περιέχει τα δεδομένα για την αρχική κατάσταση ενός παιχνιδιού λατινικών τετραγώνων.

Για παράδειγμα, εάν το όνομα του προγράμματός σας είναι `latinsolver`, και το αρχείο εισόδου είναι το αρχείο `lsq2.txt`, θα πρέπει να καλέσετε το πρόγραμμά σας με την εντολή:

```
./latinsolver lsq2.txt
```

Τα περιεχόμενα του αρχείου εισόδου έχουν την ακόλουθη μορφή:

```
4
0 0 0 -4
0 -2 0 0
0 0 0 0
0 0 0 -2
```

όπου τα δεδομένα μέσα στο αρχείο ακολουθούν την ίδια δομή όπως και στην άσκηση 1. Σημειώστε ότι οι αριθμοί της ίδιας γραμμής διαχωρίζονται με κενό (space). Κάθε γραμμή (ακόμα και η τελευταία) τερματίζεται με το χαρακτήρα αλλαγής γραμμής (enter).

Το πρόγραμμά σας θα πρέπει να τυπώνει μηνύματα στην οθόνη. Πιο συγκεκριμένα θα τυπώνετε τον πίνακα κατάστασης που εισάγεται (push) στη στοίβα και τον πίνακα κατάστασης που εξάγεται (pop) από τη στοίβα. Επίσης στο τέλος του προγράμματος θα πρέπει να τυπώνεται ο συνολικός αριθμός των push και pop που έγιναν κατά τη διάρκεια της εκτέλεσης. Δείτε παραδείγματα εκτέλεσης στο τέλος της εκφώνησης. Πρέπει να τα ακολουθήσετε πιστά.

Δώστε ιδιαίτερη προσοχή στον τρόπο κλήσης του προγράμματός σας, καθώς και στη μορφή του αρχείου εισόδου, σύμφωνα με όσα περιγράφηκαν παραπάνω (συμπεριλαμβανομένων των λύσεων που αυτό παράγει), διότι το πρόγραμμα σας ενδέχεται να ελεγχτεί με αυτόματο τρόπο πάνω σε ένα μεγάλο σύνολο δεδομένων εισόδου (black-box testing).

Με το πέρας του προγράμματος θα πρέπει να απελευθερώσετε όλες τις θέσεις μνήμης που δεσμεύτηκαν δυναμικά με χρήση malloc. Θα πρέπει να ελέγξετε το πρόγραμμά σας για διαρροή μνήμης με το πρόγραμμα Valgrind. Εκτελέστε την εντολή valgrind και **αποθηκεύστε σε αρχείο με όνομα valgrind.txt** ότι τυπώνεται στην οθόνη έτσι ώστε να λάβετε το report που δείχνει τα memory leaks, το οποίο θα πρέπει να παραδώσετε. Αυτό μπορεί να γίνει με αυτόματα – σε Unix terminal – με την εξής εντολή που ανακατευθύνει την έξοδο του valgrind σε αρχείο:

```
valgrind --tool=memcheck --leak-check=full --show-reachable=yes --num-
callers=20 --track-fds=yes --track-origins=yes ./latinsolver lsq2.txt &>
valgrind.txt
```

Για την κατασκευή της στοίβας θα πρέπει να ορίσετε μια δομή για τον κόμβο (node). Η δομή αυτή πρέπει να περιέχει τουλάχιστον τα εξής πεδία:

- Δυναμικός πίνακας ακεραίων (δείκτης σε πίνακα δύο διαστάσεων), για την αναπαράσταση της τρέχουσας κατάστασης του παιχνιδιού
- Αριθμός γραμμής (row) της θέσης που θα τοποθετηθεί ένας αριθμός
- Αριθμός στήλης (col) της θέσης που θα τοποθετηθεί ένας αριθμός
- δείκτης προς τον επόμενο κόμβο της στοίβας

Ένα υπόδειγμα κατασκευής της παραπάνω δομής ακολουθεί:

```
typedef struct node
{
    int **square;
    int row;
    int col;
    struct node *next;
} NODE;
```

Για τη στοίβα θα πρέπει να ορίσετε μια δομή που θα κρατά τον πρώτο κόμβο της στοίβας και το μέγεθος (αριθμός κόμβων) της στοίβας:

```
typedef struct
{
    NODE *top;
    int length;
} STACK;
```

Από κει και πέρα, το πρόγραμμά σας θα πρέπει να είναι ικανό μεταξύ άλλων να εκτελέσει τις εξής λειτουργίες:

- Εισαγωγή κόμβου στη στοίβα.
- Εξαγωγή κόμβου από στοίβα.
- Απελευθέρωση μνήμης

V. Γενικές Οδηγίες

Το πρόγραμμα σας θα πρέπει να συμβαδίζει με το πρότυπο ISO 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. **Να χρησιμοποιήσετε το λογισμικό τεκμηρίωσης doxygen** έτσι ώστε να μπορούμε να μετατρέψουμε τα σχόλια του προγράμματός σας σε HTML αρχεία και να τα δούμε με ένα browser. Η συστηματική αντιμετώπιση της λύσης ενός προβλήματος περιλαμβάνει στο παρόν στάδιο τη διάσπαση του προβλήματος σε μικρότερα ανεξάρτητα προβλήματα που κατά κανόνα κωδικοποιούμε σε ξεχωριστές συναρτήσεις. Για αυτό τον λόγο σας καλούμε να κάνετε χρήση συναρτήσεων και άλλων τεχνικών δομημένου προγραμματισμού που διδαχτήκατε στο ΕΠΛ131. Επίσης, σας θυμίσουμε ότι κατά την διάρκεια της εκτέλεσης του προγράμματος σας αυτό θα πρέπει να δίνει τα κατάλληλα μηνύματα σε περίπτωση λάθους. Τέλος το πρόγραμμα σας θα πρέπει να μεταγλωττίζεται στις μηχανές του εργαστηρίου. **Παρακαλώ όπως μελετηθούν ξανά οι κανόνες υποβολής εργασιών όπως αυτοί ορίζονται στο συμβόλαιο του μαθήματος.** Επίσης να ακολουθήσετε τα πιο κάτω βήματα όταν υποβάλετε την άσκηση σας στο Moodle:

Ανεβάστε στο Moodle (μόνο ένα μέλος της ομάδας να ανεβάσει) όλα τα αρχεία της εργασίας (αρχεία .c και .h με doxygen σχόλια, doxygen configuration file (να ενεργοποιηθεί και το διάγραμμα εξαρτήσεων αντικειμένων), README.md, makefile, valgrind.txt) που πρέπει να υποβάλετε ξεχωριστά. ΜΗΝ ΑΝΕΒΑΣΕΤΕ ZIP FILE.

VI. Κριτήρια αξιολόγησης

Υλοποίηση και εκτέλεση προγράμματος επίλυσης του παιχνιδιού λατινικών τετραγώνων	60
Μηνύματα εκτύπωσης (δείτε παραδείγματα εκτέλεσης)	8
Διάσπαση Προγράμματος σε Πολλαπλά Αρχεία με driver functions, function prototypes	10
Γενική εικόνα (στοιχισμένος και ευανάγνωστος κώδικας, εύστοχα ονόματα μεταβλητών, σταθερών και συναρτήσεων, README file, doxygen file, σχόλια doxygen, τήρηση των κανόνων υποβολής),	14
Valgrind report (zero definitely & indirectly memory leaks) – valgrind.txt	8
ΣΥΝΟΛΟ	100

VII. Παραδείγματα εκτέλεσης

```
>./latinsolver lsq0.txt
```

```
PUSH: STEP 1
+-----+-----+-----+
|  1  | (2) |  0  |
+-----+-----+-----+
| (2) |  0  |  0  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
PUSH: STEP 2
+-----+-----+-----+
|  1  | (2) |  3  |
+-----+-----+-----+
| (2) |  0  |  0  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
PUSH: STEP 3
+-----+-----+-----+
|  1  | (2) |  3  |
+-----+-----+-----+
| (2) |  1  |  0  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
POP: STEP 4
+-----+-----+-----+
|  1  | (2) |  3  |
+-----+-----+-----+
| (2) |  1  |  0  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
PUSH: STEP 5
+-----+-----+-----+
|  1  | (2) |  3  |
+-----+-----+-----+
| (2) |  3  |  0  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
PUSH: STEP 6
+-----+-----+-----+
|  1  | (2) |  3  |
+-----+-----+-----+
| (2) |  3  |  1  |
+-----+-----+-----+
|  0  |  0  |  0  |
+-----+-----+-----+
```

PUSH: STEP 7

```
+-----+-----+-----+
|  1  | (2) |   3  |
+-----+-----+-----+
| (2) |   3  |   1  |
+-----+-----+-----+
|  3  |   0  |   0  |
+-----+-----+-----+
```

PUSH: STEP 8

```
+-----+-----+-----+
|  1  | (2) |   3  |
+-----+-----+-----+
| (2) |   3  |   1  |
+-----+-----+-----+
|  3  |   1  |   0  |
+-----+-----+-----+
```

PUSH: STEP 9

```
+-----+-----+-----+
|  1  | (2) |   3  |
+-----+-----+-----+
| (2) |   3  |   1  |
+-----+-----+-----+
|  3  |   1  |   2  |
+-----+-----+-----+
```

PUSH NUM: 8

POP NUM: 1

>./latinsolver lsq2.txt

PUSH: STEP 1

```
+-----+-----+-----+-----+
|  1  |   0  |   0  | (4) |
+-----+-----+-----+-----+
|  0  | (2) |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  | (2) |
+-----+-----+-----+-----+
```

PUSH: STEP 2

```
+-----+-----+-----+-----+
|  1  |   3  |   0  | (4) |
+-----+-----+-----+-----+
|  0  | (2) |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  | (2) |
+-----+-----+-----+-----+
```

PUSH: STEP 3

```
+-----+-----+-----+-----+
|  1  |   3  |   2  | (4) |
+-----+-----+-----+-----+
|  0  | (2) |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  | (2) |
+-----+-----+-----+-----+
```

PUSH: STEP 4

```
+-----+-----+-----+-----+
|  1  |   3  |   2  | (4) |
+-----+-----+-----+-----+
|  3  | (2) |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  |   0  |
+-----+-----+-----+-----+
|  0  |   0  |   0  | (2) |
+-----+-----+-----+-----+
```

PUSH: STEP 5

```
+-----+-----+-----+-----+
```



```

| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 1 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

POP: STEP 6

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 1 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

PUSH: STEP 7

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 4 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

PUSH: STEP 8

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 4 | 1 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

PUSH: STEP 9

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 4 | 1 |
+---+---+---+---+
| 2 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

PUSH: STEP 10

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 4 | 1 |
+---+---+---+---+
| 2 | 1 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

PUSH: STEP 11

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |
+---+---+---+---+
| 3 | (2) | 4 | 1 |
+---+---+---+---+
| 2 | 1 | 3 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | (2) |
+---+---+---+---+

```

POP: STEP 12

```

+---+---+---+---+
| 1 | 3 | 2 | (4) |

```

```

+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 1 | 3 | 0 |
+-----+-----+-----+-----+
| 0 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

POP: STEP 13

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 1 | 0 | 0 |
+-----+-----+-----+-----+
| 0 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 14

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 4 | 0 | 0 |
+-----+-----+-----+-----+
| 0 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 15

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 4 | 1 | 0 |
+-----+-----+-----+-----+
| 0 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 16

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 4 | 1 | 3 |
+-----+-----+-----+-----+
| 0 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 17

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 4 | 1 | 3 |
+-----+-----+-----+-----+
| 4 | 0 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 18

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+
| 3 | (2) | 4 | 1 |
+-----+-----+-----+-----+
| 2 | 4 | 1 | 3 |
+-----+-----+-----+-----+
| 4 | 1 | 0 | (2) |
+-----+-----+-----+-----+

```

PUSH: STEP 19

```

+-----+-----+-----+-----+
| 1 | 3 | 2 | (4) |
+-----+-----+-----+-----+

```

```

| 3 | (2) | 4 | 1 |
+---+---+---+---+
| 2 | 4 | 1 | 3 |
+---+---+---+---+
| 4 | 1 | 3 | (2) |
+---+---+---+---+
PUSH NUM: 16
POP  NUM: 3

```

>./latinsolver lsq1.txt

```

PUSH: STEP 1
+---+---+---+---+
| 1 | (2) | 0 | (4) |
+---+---+---+---+
| (2) | (3) | 0 | 0 |
+---+---+---+---+
| 0 | 0 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+
PUSH: STEP 2
+---+---+---+---+
| 1 | (2) | 3 | (4) |
+---+---+---+---+
| (2) | (3) | 0 | 0 |
+---+---+---+---+
| 0 | 0 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+
PUSH: STEP 3
+---+---+---+---+
| 1 | (2) | 3 | (4) |
+---+---+---+---+
| (2) | (3) | 4 | 0 |
+---+---+---+---+
| 0 | 0 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+
PUSH: STEP 4
+---+---+---+---+
| 1 | (2) | 3 | (4) |
+---+---+---+---+
| (2) | (3) | 4 | 1 |
+---+---+---+---+
| 0 | 0 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+
PUSH: STEP 5
+---+---+---+---+
| 1 | (2) | 3 | (4) |
+---+---+---+---+
| (2) | (3) | 4 | 1 |
+---+---+---+---+
| 3 | 0 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+
PUSH: STEP 6
+---+---+---+---+
| 1 | (2) | 3 | (4) |
+---+---+---+---+
| (2) | (3) | 4 | 1 |
+---+---+---+---+
| 3 | 4 | (1) | (2) |
+---+---+---+---+
| 0 | 0 | 0 | (3) |
+---+---+---+---+

```

PUSH: STEP 7

```

+-----+-----+-----+-----+
|  1  | (2) |  3  | (4) |
+-----+-----+-----+-----+
| (2) | (3) |  4  |  1  |
+-----+-----+-----+-----+
|  3  |  4  | (1) | (2) |
+-----+-----+-----+-----+
|  4  |  0  |  0  | (3) |
+-----+-----+-----+-----+

```

PUSH: STEP 8

```

+-----+-----+-----+-----+
|  1  | (2) |  3  | (4) |
+-----+-----+-----+-----+
| (2) | (3) |  4  |  1  |
+-----+-----+-----+-----+
|  3  |  4  | (1) | (2) |
+-----+-----+-----+-----+
|  4  |  1  |  0  | (3) |
+-----+-----+-----+-----+

```

PUSH: STEP 9

```

+-----+-----+-----+-----+
|  1  | (2) |  3  | (4) |
+-----+-----+-----+-----+
| (2) | (3) |  4  |  1  |
+-----+-----+-----+-----+
|  3  |  4  | (1) | (2) |
+-----+-----+-----+-----+
|  4  |  1  |  2  | (3) |
+-----+-----+-----+-----+

```

PUSH NUM: 9

POP NUM: 0

>./latinsolver lsq3.txt

PUSH: STEP 1

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|  2  |  0  |  0  | (4) |  0  | (6) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  | (2) |  0  |  0  | (1) | (7) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  | (1) |  0  |  0  |  0  | (3) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  |  0  |  0  | (2) |  0  |  0  |  0  |
+-----+-----+-----+-----+-----+-----+-----+
| (6) | (4) |  0  |  0  |  0  |  0  | (1) |
+-----+-----+-----+-----+-----+-----+-----+
|  0  |  0  |  0  |  0  |  0  |  0  |  0  |
+-----+-----+-----+-----+-----+-----+-----+
| (1) |  0  |  0  |  0  | (5) |  0  | (7) |
+-----+-----+-----+-----+-----+-----+-----+

```

PUSH: STEP 2

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|  2  |  3  |  0  | (4) |  0  | (6) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  | (2) |  0  |  0  | (1) | (7) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  | (1) |  0  |  0  |  0  | (3) |  0  |
+-----+-----+-----+-----+-----+-----+-----+
|  0  |  0  |  0  | (2) |  0  |  0  |  0  |
+-----+-----+-----+-----+-----+-----+-----+
| (6) | (4) |  0  |  0  |  0  |  0  | (1) |
+-----+-----+-----+-----+-----+-----+-----+
|  0  |  0  |  0  |  0  |  0  |  0  |  0  |
+-----+-----+-----+-----+-----+-----+-----+
| (1) |  0  |  0  |  0  | (5) |  0  | (7) |
+-----+-----+-----+-----+-----+-----+-----+

```

... (δεν φαίνονται όλα τα βήματα λόγω του μεγάλου όγκου αποτελεσμάτων)

PUSH: STEP 1220

```

+-----+-----+-----+-----+-----+-----+
| 2 | 3 | 1 | (4) | 7 | (6) | 5 |
+-----+-----+-----+-----+-----+-----+
| 3 | (2) | 4 | 5 | (1) | (7) | 6 |
+-----+-----+-----+-----+-----+-----+
| 5 | (1) | 7 | 6 | 2 | (3) | 4 |
+-----+-----+-----+-----+-----+-----+
| 7 | 5 | 6 | (2) | 4 | 1 | 3 |
+-----+-----+-----+-----+-----+-----+
| (6) | (4) | 5 | 7 | 3 | 2 | (1) |
+-----+-----+-----+-----+-----+-----+
| 4 | 7 | 3 | 1 | 6 | 5 | 2 |
+-----+-----+-----+-----+-----+-----+
| (1) | 6 | 0 | 0 | (5) | 0 | (7) |
+-----+-----+-----+-----+-----+-----+

```

PUSH: STEP 1221

```

+-----+-----+-----+-----+-----+-----+
| 2 | 3 | 1 | (4) | 7 | (6) | 5 |
+-----+-----+-----+-----+-----+-----+
| 3 | (2) | 4 | 5 | (1) | (7) | 6 |
+-----+-----+-----+-----+-----+-----+
| 5 | (1) | 7 | 6 | 2 | (3) | 4 |
+-----+-----+-----+-----+-----+-----+
| 7 | 5 | 6 | (2) | 4 | 1 | 3 |
+-----+-----+-----+-----+-----+-----+
| (6) | (4) | 5 | 7 | 3 | 2 | (1) |
+-----+-----+-----+-----+-----+-----+
| 4 | 7 | 3 | 1 | 6 | 5 | 2 |
+-----+-----+-----+-----+-----+-----+
| (1) | 6 | 2 | 0 | (5) | 0 | (7) |
+-----+-----+-----+-----+-----+-----+

```

PUSH: STEP 1222

```

+-----+-----+-----+-----+-----+-----+
| 2 | 3 | 1 | (4) | 7 | (6) | 5 |
+-----+-----+-----+-----+-----+-----+
| 3 | (2) | 4 | 5 | (1) | (7) | 6 |
+-----+-----+-----+-----+-----+-----+
| 5 | (1) | 7 | 6 | 2 | (3) | 4 |
+-----+-----+-----+-----+-----+-----+
| 7 | 5 | 6 | (2) | 4 | 1 | 3 |
+-----+-----+-----+-----+-----+-----+
| (6) | (4) | 5 | 7 | 3 | 2 | (1) |
+-----+-----+-----+-----+-----+-----+
| 4 | 7 | 3 | 1 | 6 | 5 | 2 |
+-----+-----+-----+-----+-----+-----+
| (1) | 6 | 2 | 3 | (5) | 0 | (7) |
+-----+-----+-----+-----+-----+-----+

```

PUSH: STEP 1223

```

+-----+-----+-----+-----+-----+-----+
| 2 | 3 | 1 | (4) | 7 | (6) | 5 |
+-----+-----+-----+-----+-----+-----+
| 3 | (2) | 4 | 5 | (1) | (7) | 6 |
+-----+-----+-----+-----+-----+-----+
| 5 | (1) | 7 | 6 | 2 | (3) | 4 |
+-----+-----+-----+-----+-----+-----+
| 7 | 5 | 6 | (2) | 4 | 1 | 3 |
+-----+-----+-----+-----+-----+-----+
| (6) | (4) | 5 | 7 | 3 | 2 | (1) |
+-----+-----+-----+-----+-----+-----+
| 4 | 7 | 3 | 1 | 6 | 5 | 2 |
+-----+-----+-----+-----+-----+-----+
| (1) | 6 | 2 | 3 | (5) | 4 | (7) |
+-----+-----+-----+-----+-----+-----+

```

PUSH NUM: 629

POP NUM: 594

Καλή Επιτυχία!