

HY – 100: Εισαγωγή στην Επιστήμη Υπολογιστών

4η Σειρά Ασκήσεων

Οδηγίες

Για τη μεταγλώττιση των προγραμμάτων που ζητούνται θα πρέπει να χρησιμοποιήσετε τον gcc με τις παρακάτω παραμέτρους:

```
gcc -ansi -pedantic -Werror
```

ΠΡΟΣΟΧΗ: Οι ασκήσεις σας βαθμολογούνται με αυτόματο τρόπο, οπότε θα πρέπει να υπακούν πιστά στην εκφώνηση (ακόμη και τα ονόματα των αρχείων). Για να δοκιμάσετε την άσκηση σας αντιγράψτε το αρχείο hw4-tests.tgz στον φάκελο με την άσκησή σας με την εντολή:

```
cp ~hy100/public_html/hw4-tests.tgz ./
```

η κατεβάστε το από το <http://www.csd.uoc.gr/~hy100/hw4-tests.tgz>. Αποσυμπιέστε το στον φάκελο με την άσκηση σας, εκτελώντας την εντολή:

```
tar xvf hw4-tests.tgz
```

Τρέξτε την άσκηση σας με τα αυτόματα tests εκτελώντας την εντολή:

```
sh test.sh ./hw4 tests
```

όπου ./hw4 είναι το όνομα του προγράμματός σας και tests ο φάκελος με τα tests. Κάθε test αποτελείται από ένα αρχείο με κατάληξη .in, που περιέχει την είσοδο για το πρόγραμμά σας. Αντίστοιχα, το .out αρχείο που περιέχει την αναμενόμενη έξοδο του προγράμματος.

Όταν ένα test εκτελείται με επιτυχία τυπώνεται PASS, αλλιώς FAIL.

1. Μαγικό Τετράγωνο

Μαγικό τετράγωνο είναι ένα $N \times N$ τετράγωνο το οποίο έχει τα εξής χαρακτηριστικά:

- Περιέχει όλους τους αριθμούς από 1 έως N^2 .
- Το άθροισμα κάθε γραμμής, κάθε στήλης και κάθε μιας από τις δύο διαγώνιους είναι το ίδιο και ίσο με $N \times (N^2 + 1) / 2$

Ενα παράδειγμα 3×3 μαγικού τετραγώνου είναι η εικόνα στα δεξιά.

Σκοπός της άσκησης είναι να γράψετε ένα πρόγραμμα το οποίο θα μπορεί να ελέγχει εάν ένας $N \times N$ πίνακας είναι μαγικό τετράγωνο. Μπορείτε να υποθέσετε ότι $N < 100$, το οποίο σημαίνει ότι στην αρχή του προγράμματός σας θα πρέπει να ορίσετε ότι:

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15

```
#define MAXN 100
```

Το πρόγραμμα θα πρέπει να διαβάζει ένα πίνακα από το πληκτρολόγιο (χρησιμοποιώντας τη scanf). Ο πρώτος ακέραιος που διαβάζει είναι το N (η διάσταση του πίνακα). Στη συνέχεια, το πρόγραμμα πρέπει να διαβάζει κάθε ένα από τα στοιχεία του πίνακα (συνολικά N^2).

Παράδειγμα εισόδου:

3		
2	7	6
9	5	1
4	3	8

Το πρόγραμμά σας, αφού διαβάσει τον πίνακα, θα πρέπει να εξετάζει όλες τις γραμμές, στήλες, και τις δύο διαγώνιους, καθώς και το εάν τα στοιχεία του πίνακα είναι μοναδικά (δηλαδή περιέχει ακριβώς μία φορά τους αριθμούς 1 έως N^2).

Για κάθε ένα από τα παραπάνω θα πρέπει να τυπώνει το είδος του ελέγχου (ROW, COLUMN, DIAG1, DIAG2, UNIQUE) και να τυπώνει YES ή NO αναλόγως αν ο έλεγχος πέτυχε ή απέτυχε. Στο τέλος, θα αναφέρει αν το τετράγωνο είναι όντως μαγικό ή όχι, τυπώνοντας MAGIC YES ή MAGIC NO αντίστοιχα. Για παράδειγμα αν δώσουμε την παραπάνω είσοδο, η έξοδος θα πρέπει να είναι:

ROW 1 YES

ROW 2 YES

ROW 3 YES

COLUMN 1 YES

COLUMN 2 YES

COLUMN 3 YES

DIAG1 YES

DIAG2 YES

UNIQUE YES

MAGIC YES

Οπου DIAG1 είναι η διαγώνιος που αρχίζει άνω αριστερά και DIAG2 η διαγώνιος που αρχίζει πάνω δεξιά.

Αντίθετα, αν δώσουμε σαν είσοδο:

3		
2	7	1
9	5	1
4	3	8

Η έξοδος θα πρέπει να είναι:

ROW 1 NO
 ROW 2 YES
 ROW 3 YES
 COLUMN 1 YES
 COLUMN 2 YES
 COLUMN 3 NO
 DIAG1 YES
 DIAG2 NO
 UNIQUE NO
 MAGIC NO

Θα πρέπει να ελέγχετε ότι οι αριθμοί είναι μεταξύ του διαστήματος από 1 έως και N^2 .

2. Run-length Encoding

Ο Run-length encoding είναι ένας από τους πιο απλούς αλγόριθμους συμπίεσης δεδομένων χωρίς απώλειες. Με είσοδο ένα αρχείο οποιουδήποτε τύπου, κωδικοποιεί τα δεδομένα αντικαθιστώντας κάθε ακολουθία όμοιων bytes με το πλήθος τους και την κοινή τους τιμή. Για παράδειγμα, έστω η ακολουθία χαρακτήρων:

WWWWBWWWWWWWWWWWWWWBBBWWWWWWWWWWBWWWWWWWWWWWWWWWW

Τότε, ο αλγόριθμος του run-length encoding θα επιστρέψει την ακολουθία

4W1B12W3B9W1B13W

Σε αυτή την άσκηση, καλείστε να υλοποιήσετε ένα πρόγραμμα που παρέχει δύο λειτουργίες με βάση τον παραπάνω αλγόριθμο:

- **Κωδικοποίηση (encoding):** Σε αυτή τη λειτουργία, το πρόγραμμα σας λαμβάνει ως είσοδο από το πληκτρολόγιο μία συμβολοσειρά που αποτελείται μόνο από χαρακτήρες του λατινικού αλφαβήτου(a-z, A-Z) και ως έξοδο παράγει το αποτέλεσμα του run-length encoding αλγορίθμου
- **Αποκωδικοποίηση (decoding):** Αυτή η λειτουργία αποτελεί το αντίστροφο της προηγούμενης. Το πρόγραμμα σας λαμβάνει ως είσοδο από το πληκτρολόγιο το

αποτέλεσμα της λειτουργίας encoding και παράγει ως έξοδο την αρχική συμβολοσειρά λατινικών χαρακτήρων.

Η επιλογή των λειτουργιών θα γίνεται χρησιμοποιώντας μία παράμετρο γραμμής εντολών, όπου όταν αυτή η παράμετρος είναι 'e' εκτελείται η λειτουργία κωδικοποίησης ενώ όταν είναι 'd' εκτελείται η λειτουργία αποκωδικοποίησης. Μπορείτε να υποθέσετε ότι το πρόγραμμά σας θα καλείται πάντα με την κατάλληλη παράμετρο γραμμής εντολών, καθώς και ότι η είσοδος που δίνεται είναι ορθή. Δηλαδή, για την λειτουργία κωδικοποίησης η είσοδος είναι πάντα συμβολοσειρά που περιέχει μόνο λατινικούς χαρακτήρες ενώ κατά την αποκωδικοποίηση η είσοδος αποτελείται μόνο από «ζευγάρια» ακεραίου - λατινικού χαρακτήρα.

Παραδείγματα εκτελέσεων:

```
$ ./run_length e
WWWWBWWWWWWWWWWWWBWWWWWWWWWWWWWWWWWW
4W1B12W3B9W1B13W
```

```
$ ./run_length d
4W1B12W3B9W1B13W
WWWWBWWWWWWWWWWWWBWWWWWWWWWWWWWWWWWW
```

Συναρτήσεις που ίσως χρησιμοποιήσετε: getchar, putchar, scanf, sscanf, printf, strcmp

Για να τρέξετε τα αυτοματοποιημένα τεστ αυτής της άσκησης χρησιμοποιήστε τις ακόλουθες εντολές (προσέξτε τα εισαγωγικά που χρησιμοποιούνται):

Για την λειτουργία κωδικοποίησης:

```
$ sh test.sh "hw4b e" tests2.1
```

Για την λειτουργία αποκωδικοποίησης:

```
$ sh test.sh "hw4b d" tests2.2
```

Οδηγίες Παράδοσης Σειράς Ασκήσεων

Παραδώστε τα αρχεία hw4a.c και hw4b.c ακολουθώντας τις οδηγίες που βρίσκονται στη σελίδα <https://www.csd.uoc.gr/~hy100/turnin-howto-gr.html>

Κάθε Επιτυχία !