Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
http://mycourses.ntua.gr

Προγραμματιστικές Τεχνικές

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 1

*Οι ασκήσεις της σειράς αυτής αποτελούν μέρος (και συνέχεια) της εργαστηριακής εξάσκησης στο μάθημα "Προγραμματιστικές Τεχνικές". Η υποβολή του σχετικού προγράμματος έμμεσα δηλώνει ότι **είσθε οι μοναδικοί συγγραφείς της λύσης της άσκησης**. Εάν το πρόγραμμα ή μέρος του έχει αντιγραφεί θα πρέπει να αναφέρεται η πηγή και ο συγγραφέας του.*

ΠΕΡΙΓΡΑΦΗ

Άσκηση Α1-1

Η πρώτη άσκηση έχει σαν σκοπό να σχεδιάσετε και υλοποιήσετε λειτουργίες σε απλά συνδεδεμένη λίστα. Πιο συγκεκριμένα, θα πρέπει:

α) Να ορίσετε τύπο απλά συνδεδεμένης γραμμικής λίστας, της οποίας οι κόμβοι περιέχουν ονόματα.
β) Να γράψετε τη συνάρτηση createlist, η οποία διαβάζει μια ακολουθία ονομάτων στην είσοδο από ένα αρχείο και κατασκευάζει την αντίστοιχη λίστα, τοποθετώντας τα ονόματα με τη σειρά εμφάνισής τους.
γ) Να γράψετε τη συνάρτηση printlist, η οποία εκτυπώνει τα ονόματα που περιέχονται στη λίστα με την ίδια σειρά στο stdout και σε κάποιο αρχείο εξόδου.
δ) Να γράψετε τη συνάρτηση length, η οποία υπολογίζει το πλήθος των κόμβων μιας λίστας και τυπώνει το αποτέλεσμα στο stdout.
ε) Να γράψετε τη συνάρτηση insertsublist(x, j, y, k, n), η οποία αντιγράφει στη λίστα y αρχίζοντας από τη θέση k μια υπολίστα της λίστας x μήκους n αρχίζοντας από τη θέση j. Η λίστα x παραμένει αμετάβλητη. Αν length(x)<j+n-1 ή length(y)<k-1 ή j<1 ή k<1, θα τυπώνονται κατάλληλα μηνύματα λάθους.
στ) Να γράψετε κύριο πρόγραμμα το οποίο διαβάζει δύο λίστες από ένα αρχείο εισόδου in.dat και τις εκτυπώνει στο stdout αφού εφαρμόσει καταλλήλως τη συνάρτηση insertsublist.

Το format του αρχείου εισόδου είναι:

```
List 1
<onoma>
<onoma>
<onoma>
.....
List 2
<onoma>
<onoma>
.....
<onoma>
```

Άσκηση Α1-2

Η δεύτερη άσκηση έχει σαν σκοπό τη σχεδίαση και υλοποίηση λειτουργιών σε πίνακες δύο διαστάσεων με δομές. Για βοήθεια σε αυτή την άσκηση μπορείτε να δείτε και το Παράδειγμα Τ5-4 στην ιστοσελίδα του μαθήματος. Πιο συγκεκριμένα, το αντικείμενο της άσκησης είναι να σχεδιάσετε και να υλοποιήσετε ένα πρόγραμμα που αναλύει τιμές στα κελιά ενός πίνακα δύο διαστάσεων και βρίσκει τα σημεία (κελιά), που οι τιμές από τα γειτονικά τους κελιά, διαφέρουν κατά κάποιο συγκεκριμένο ποσοστό. Αυτού του είδους οι αλγόριθμοι χρησιμοποιούνται για την αναγνώριση ακμών σε ψηφιακές εικόνες (edge detection). Αναλυτικά, το ζητούμενο πρόγραμμα θα έχει τα παρακάτω στοιχεία και δομές.

α) Ο πίνακας θα υλοποιηθεί με δομές όπως στα Παραδείγματα Τ5-3 και Τ5-4 δηλαδή:

```
typedef struct {
         int red;
         int green;
         int blue;
         int column;
} TableElementType;

typedef struct {
        int rowNumber;
        TableElementType *theContents;
} MatrixElementType;
```

β) Τα στοιχεία του πίνακα θα διαβάζονται από ένα αρχείο in.dat με format:

```
<Αριθμ. Γραμμών> <Αριθμ. Στηλών>
<τιμή-red> <τιμή-green> <τιμή-blue>
<τιμή-red> <τιμή-green> <τιμή-blue>
.....................
<τιμή-red> <τιμή-green> <τιμή-blue>
```

(τόσες τριάδες όσο το πλήθος των στηλών του πίνακα)

(τόσες γραμμές όσες το πλήθος των γραμμών του πίνακα)

γ) Οι δυνατές τιμές για τα πεδία red, green, blue είναι οι ακέραιοι αριθμοί στο εύρος 0-255.

δ) Τα πρόγραμμα θα επεξεργάζεται τα στοιχεία με μια συνάρτηση που θα λαμβάνει ως παραμέτρους τον πίνακα που πρόκειται να επεξεργαστούμε, και το ποσοστό διαφοράς τιμών που ορίζει ο χρήστης ως όριο για την αναγνώριση ακμών (πραγματικές τιμές στο διάστημα [0.0, 100.0]). Ο σκοπός είναι να κατασκευαστεί ένας νέος πίνακας ίδιων διαστάσεων με τον αρχικό όπου, εάν ο μέσος όρος των τιμών red, green, blue ενός κελιού του αρχικού διαφέρει από την αντίστοιχη τιμή του μέσου όρου κάποιου από τα γειτονικά του κελιά (αριστερά, δεξιά, πάνω, κάτω) κατά ποσοστό μεγαλύτερο μιας δεδομένης τιμής που ορίζεται από την παράμετρο, τότε το αντίστοιχο σημείο (κελί) του νέου πίνακα έχει τιμή 1 στα πεδία red, green, blue ενώ για τα άλλα σημεία (κελιά) έχει τιμή 0. Ο νέος πίνακας θα εκτυπώνεται σε ένα αρχείο εξόδου. Αυτή η βασική συνάρτηση επεξεργασίας θα έχει ως πρότυπο την MatrixElementType * processImages(MatrixElementType InputMatrιx[], MatrixElementType outMatrix[], int row, int col, float percentage) όπου η πρώτη και δεύτερη παράμετρος είναι οι πίνακες εισόδου και αποτελέσματος αντίστοιχα, η τρίτη και τέταρτη παράμετρος είναι ακέραιοι n, m που ορίζουν το μέγεθος των πινάκων (εισόδου, αποτελέσματος) σε γραμμές και στήλες αντίστοιχα, και η πέμπτη παράμετρος ορίζει το ποσοστό διαφοράς που είναι το όριο αναγνώρισης ακμών. Υλοποιήστε τη παραπάνω συνάρτηση μόνο με τη χρήση δεικτών για τη προσπέλαση των στοιχείων του διανύσματος. Η συνάρτηση επιστρέφει τη διεύθυνση του πρώτου στοιχείου του πίνακα αποτελέσματος.

ε) Η λογική ελέγχου για το έλεγχο της ποσοστιαίας διαφοράς τιμών ενός κελιού Α από ένα γειτονικό του κελί Β δίνεται από την έκφραση:

- $|c_A - c_B| / c_A * 100$ εάν το $c_A \neq 0$, και
- 100 εάν το $c_A = 0$

όπου c_A είναι ο μέσος όρος των τιμών red, green, blue του τρέχοντος κελιού και c_B ο μέσος όρος των τιμών red, green, blue ενός γειτονικού του κελιού

στ) Το κύριο πρόγραμμα θα α) διαβάζει τον πίνακα στοιχείων εισόδου, β) επεξεργάζεται τον πίνακα, γ) τυπώνει σε ένα αρχείο out.dat τον αρχικό πίνακα

και τον πίνακα αποτέλεσμα δ) τυπώνει τον αρχικό πίνακα εισόδου και τον πίνακα αποτέλεσμα στο stdout. Όσον αφορά την εκτύπωση των δύο πινάκων εκτυπώνονται μόνο οι μέσοι όροι των τιμών red, green, blue για κάθε θέση του πίνακα.

Άσκηση Α1-3

Δείτε το παραδείγματα Τ3-5 και Τ5-3 σαν πιθανά σημεία αναφοράς χρήσης πινάκων και σχεδιάστε και υλοποιήστε ένα πρόγραμμα στη γλώσσα C που να υπολογίζει και επιστρέφει τον αντίστροφο ενός τετραγωνικού πίνακα ο οποίος αποτελείται από ακέραιες τιμές. Μπορείτε να δείτε διαφορετικές μεθόδους υπολογισμού του αντίστροφου ενός πίνακα nXn στη παρακάτω ιστοσελίδα http://math.uww.edu/~mcfarlat/inverse.htm. Μπορείτε να θεωρήσετε ότι έχετε δύο πίνακες ένα για είσοδο με ακέραιες τιμές, και ένα για το αποτέλεσμα (αντίστροφος), με πραγματικές τιμές. Το πρόγραμμά σας θα διαβάζει τα στοιχεία του πίνακα ανά γραμμή από ένα αρχείο με format:

```
<Αριθμ. Γραμμών> <Αριθμ. Στηλών>
<τιμή> <τιμή> .... <τιμή>
.........................
<τιμή> <τιμή> .... <τιμή>
```

θα υπολογίζει τον αντίστροφο πίνακα, και θα εκτυπώνει τον αρχικό πίνακα και τον αντίστροφό πίνακα σε ένα αρχείο out.dat, και στο stdout.

Σχόλια - Παραδοτέα

Τα προγράμματά σας θα πρέπει να είναι **δομημένα** και να αποτελούνται από τις προγραμματιστικές **ενότητες** (modules):

α) Ένα .h αρχείο με όλες τις δηλώσεις τύπων
β) Ένα .c αρχείο με τις δηλώσεις και τις υλοποιήσεις των συναρτήσεων που ορίζουν τις λειτουργίες των προγραμμάτων καθώς και τις δηλώσεις και υλοποιήσεις της συνάρτησης main, και όποιων άλλων συναρτήσεων χρειάζονται.
γ) Καλά δομημένος κώδικας με σχόλια και καλά επιλεγμένα ονόματα μεταβλητών
δ) Τα αποτελέσματα σε ξεχωριστά αρχεία
ε) Το πακέτο αποστολής θα είναι ένα zip αρχείο που θα περιλαμβάνει όλα τα .h, .c προγράμματα και τα αρχεία δεδομένων εισόδου και αποτελεσμάτων σε διαφορετικά directories για κάθε άσκηση, καθώς και ένα επεξηγηματικό readme αρχείο με πληροφορίες σχετικές με την χρήση των προγραμμάτων που παραδίδετε.
στ) Η καταληκτική ημερομηνία και ώρα παράδοσης είναι τα μεσάνυκτα της Τετάρτης 10 Ιουνίου.
στ) Οι οδηγίες αποστολής θα αναρτηθούν στην ιστοσελίδα του μαθήματος.