



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ ΤΜΗΜΑ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΝΕΜΗΜΕΝΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
<http://courses.ece.tuc.gr>

ΗΜΜΥ ΠΛΗ 101

ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΗΛΕΚΤΡΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΕΣ ΚΑΙ ΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ: 14.02.2012

ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ: ΘΑ ΑΝΑΚΟΙΝΩΘΕΙ

ΥΠΕΥΘΥΝΟΙ ΕΡΓΑΣΤΗΡΙΟΥ: Ανέστης Γιώργος, Καζάσης Φώτης, Μουμουτζής Νεκτάριος
(ganest, fotis, nektar) @ced.tuc.gr

ΒΟΗΘΟΙ ΜΑΘΗΜΑΤΟΣ: Μανώλης Μυλωνάκης
(manolis) @ced.tuc.gr

Δεύτερος Κύκλος Ασκήσεων

Οι παρακάτω ασκήσεις είναι μέρος των εργαστηρίων του μαθήματος ΗΜΜΥ 101, και είναι υποχρεωτικές. Ο δεύτερος κύκλος αντιστοιχεί στο 60% του συνολικού βαθμού του εργαστηρίου. Οι ασκήσεις γίνονται **ατομικά**.

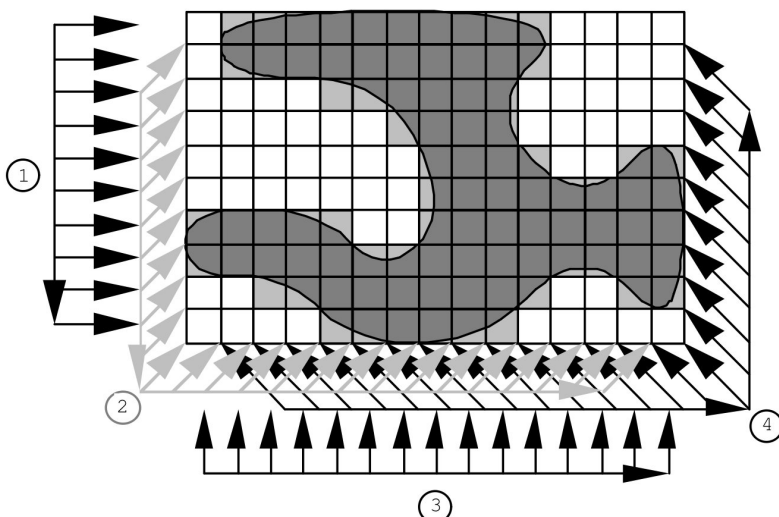
Ο πρώτος κύκλος ασκήσεων πρέπει να σταλεί, σύμφωνα με τις οδηγίες που δίνονται παρακάτω, μέχρι τις **15:00 η ώρα της 14ης Φεβρουαρίου 2012**.

Η υποβολή του προγράμματος αυτόματα δηλώνει ότι είστε οι μοναδικοί συγγραφείς της λύσης της άσκησης. Εάν το πρόγραμμα ή μέρος αυτού έχει αντιγραφεί, θα πρέπει να αναφέρεται η πηγή του και ο συγγραφέας του. Τα μέρη της λύσης που έχουν αντιγραφεί δε θα βαθμολογούνται αναλογικά.

ΠΡΩΤΗ ΑΣΚΗΣΗ (50%)

Περιγραφή

Ένας σαρωτής σώματος λειτουργεί με τη βοήθεια αισθητήρων που ανιχνεύουν την πυκνότητα μάζας κατά μήκος γραμμικών αξόνων. Στη συνέχεια, γίνεται επεξεργασία των τιμών που κατέγραψαν οι αισθητήρες ώστε να παραχθεί μια εικόνα που αναπαριστά το σώμα που σαρώθηκε. Αντικείμενο αυτής της άσκησης είναι η κατασκευή ενός προγράμματος το οποίο θα κάνει την ανωτέρω επεξεργασία. Το πρόγραμμα θα δέχεται ως είσοδο ένα αρχείο με τις τιμές που κατέγραψαν οι αισθητήρες και θα παράγει ως έξοδο ένα άλλο αρχείο το οποίο αντιστοιχεί στην εικόνα που προέκυψε από την επεξεργασία.



Θεωρείστε ότι ο σαρωτής αποτελείται από τέσσερις σειρές αισθητήρων τοποθετημένους γύρω από ένα πλέγμα διαστάσεων $N \times M$, όπως φαίνεται στο σχήμα παραπλεύρως, όπου τα N και M έχουν τιμές 10 και 15 αντίστοιχα. Η σειρά αισθητήρων (1) αποτελείται από 10 αισθητήρες που "βλέπουν" οριζόντια από αριστερά προς τα δεξιά. Η σειρά (2) έχει 24 αισθητήρες που "βλέπουν" διαγώνια με κατεύθυνση από κάτω αριστερά προς πάνω δεξιά. Η σειρά (3) έχει 15 αισθητήρες που "βλέπουν" κάθετα από κάτω προς τα πάνω. Τέλος, η σειρά (4) έχει 24 αισθητήρες που "βλέπουν" διαγώνια από κάτω δεξιά προς πάνω αριστερά.

Κατά τη διάρκεια μιας σάρωσης ο κάθε αισθητήρας καταγράφει την πυκνότητα του αντικειμένου κατά μήκος της αντίστοιχης ευθείας πάνω στο πλέγμα. Συγκεκριμένα, ο κάθε αισθητήρας σαρώνει μια "ακτίνα" και ουσιαστικά μετρά σε πόσα κελιά του πλέγματος η πυκνότητα είναι μεγάλη (τιμή 1), και σε πόσα είναι μικρή (τιμή 0), δηλαδή αθροίζει τους άσσους κατά μήκος της γραμμής σάρωσης της μαυρόασπρης εικόνας. Επομένως, οι τιμές που προκύπτουν μετά από μια σάρωση είναι τόσες όσες και οι "ακτίνες" σάρωσης. Η ανάγνωση των τιμών που δίνουν οι αισθητήρες γίνεται με φορά αντίστροφη από τη φορά κίνησης των δεικτών του ρολογιού. Μια πλήρης σάρωση αποτελείται από 73 τιμές.

Κατασκευάστε ένα πρόγραμμα στη γλώσσα C το οποίο θα διαβάζει ένα αρχείο που περιέχει τις τιμές που κατέγραψαν οι αισθητήρες, θα παράγει την εικόνα που αντιστοιχεί στις τιμές εισόδου και θα την εκτυπώνει σε ένα αρχείο.

ΣΗΜΕΙΩΣΗ: Στη γενική μορφή του προβλήματος ενδέχεται να υπάρχουν περιπτώσεις ασάφειας όπου οι τιμές των αισθητήρων δεν αρκούν για την ανακατασκευή της εικόνας. Θεωρείστε ότι το πρόγραμμά σας **δεν** χρειάζεται να ασχοληθεί με την αντιμετώπιση αυτών των περιπτώσεων.

Αρχείο εισόδου

Το αρχείο εισόδου ξεκινά με δύο αριθμούς που αντιστοιχούν στις τιμές των N και M. Στη συνέχεια περιέχει τις τιμές που κατέγραψαν οι αισθητήρες της σειράς (1), μετά της σειράς (2), της σειράς (3) και της σειράς (4) στο τέλος. Για το παράδειγμα της εικόνας του προηγούμενου σχήματος, το αρχείο εισόδου είναι το εξής:

```
10 15
10 10 6 4 6 8 13 15 11 6
0 1 2 2 2 2 4 5 5 6 7 6 5 6 6 5 5 6 6 3 2 2 1 0
2 4 5 5 7 6 7 10 10 10 7 3 3 5 5
0 0 1 3 4 4 4 4 3 4 5 7 8 8 9 9 6 4 4 2 0 0 0 0
```

Αρχείο εξόδου

Το αρχείο εξόδου θα αντιστοιχεί σε μια ασπρόμαυρη εικόνα. Τα εικονοστοιχεία που αναπαριστούν περιοχές σώματος (σκούρα στο παράδειγμα του σχήματος) εμφανίζονται με το χαρακτήρα '#'. Τα υπόλοιπα εικονοστοιχεία (λευκά στο παράδειγμα του σχήματος) εμφανίζονται με το χαρακτήρα '.'.

Για την εικόνα του παραδείγματος το αρχείο εξόδου θα έχει τη μορφή:

```
#####....
#####....
...#####....
.....####....
.....#####.##
.....#####
#####.#####
#####
..#####.##
...#####....
```

ΠΡΟΣΟΧΗ: Το πρόγραμμά σας θα πρέπει να εκτελεί όλους τους απαραίτητους ελέγχους που θα εξασφαλίζουν τη σωστή λειτουργία του.

Τι θα πρέπει να παραδώσετε

Το αρχείο με τον πηγαίο κώδικά σας με όνομα της μορφής x_set2_exer1.c, , όπου x θα το αντικαταστήσετε με τον κωδικό της ομάδας σας (π.χ. LAB101030_set2_exer1.c). Προσοχή στη στοίχιση του κώδικα και τα σχόλια, τα οποία θα πρέπει να είναι επεξηγηματικά του κώδικα. Στις πρώτες γραμμές του πηγαίου κώδικα να αναφέρετε το όνομά σας και τον αριθμό του φοιτητικού μητρώου σας.

ΔΕΥΤΕΡΗ ΑΣΚΗΣΗ (50%)

Περιγραφή

Σχεδιάστε και υλοποιήσετε ένα πρόγραμμα στη γλώσσα προγραμματισμού C το οποίο θα διαχειρίζεται μία μικρή βάση δεδομένων με τους μαθητές μίας τάξης και τις επιδόσεις τους στα μαθήματα. Ο κάθε μαθητής θα μοντελοποιείται με μια δομή (struct) στη παρακάτω μορφή:

```
typedef struct {
    char * firstName;
    char * lastName;
    int idNumber;
    StudentRecordType marks;
} StudentType;
```

Ο «φάκελος» κάθε μαθητή με τους βαθμούς της πρώτης και δεύτερης άσκησης, της προόδου, και του τελικού διαγωνίσματος μοντελοποιείται σε μία δομή όπως παρακάτω:

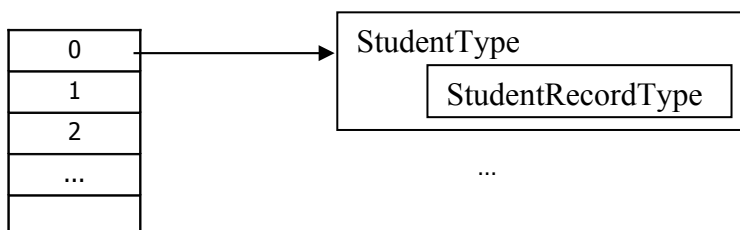
```
typedef struct {
    float firstAssignment; /* Ο βαθμός της πρώτης άσκησης */
    float secondAssignment; /* Ο βαθμός της δεύτερης άσκησης */
    float midterm; /* Ο βαθμός της προόδου */
    float final; /* Ο βαθμός του τελικού διαγωνίσματος */
    float finalMark; /* Ο γενικός βαθμός για το μάθημα */
} StudentRecordType;
```

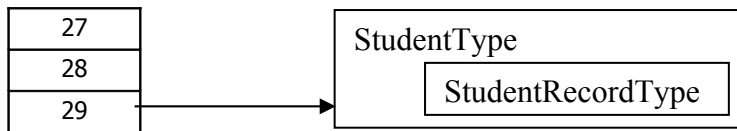
Υπολογίστε ότι γι' αυτή την ιδεατή τάξη η πρώτη άσκηση είναι το 10% του βαθμού, η δεύτερη άσκηση το 15% του βαθμού, η πρόοδος το 25% του βαθμού, και το τελικό διαγώνισμα το 50% του βαθμού. Όλες οι βαθμολογίες μη αρνητικές με άριστα το 100. Υποθέστε ότι η τάξη έχει το πολύ 30 μαθητές.

Το πρόγραμμά σας θα διαβάξει ένα αρχείο με όνομα `project2-askhsh2.dat` της μορφής:

<πλήθος μαθητών>		10
<όνομα> <επώνυμο>		Yiannis Yiannopoulos
<βαθμός πρώτης άσκησης>		90
<βαθμός δεύτερης άσκησης>		78
<βαθμός προόδου>		67
<βαθμός τελικού διαγωνίσματος>		80
<όνομα> <επώνυμο>	Π.Χ.	Kostas Kontogiannis
<βαθμός πρώτης άσκησης>		50
<βαθμός δεύτερης άσκησης>		60
<βαθμός προόδου>		40
<βαθμός τελικού διαγωνίσματος>		90
.....	
END		END

Γραφικά η δομή της τάξης είναι όπως στο σχήμα που ακολουθεί:





Να υλοποιήσετε ένα πρόγραμμα το οποίο:

1. Θα διαβάζει τα δεδομένα
2. Θα υπολογίζει τον τελικό βαθμό κάθε μαθητή
3. Θα υπολογίζει το μέσο όρο της τάξης
4. Θα βρίσκει και θα τυπώνει τον φάκελο κάποιου μαθητή βάσει του επωνύμου του
5. Θα τυπώνει αλφαβητικά τα αποτελέσματα με τη παρακάτω μορφή:

ΑΠΟΤΕΛΕΣΜΑΤΑ

ΑΝΑΛΥΤΙΚΑ

<όνομα> <επώνυμο> <τελικός βαθμός>

<όνομα> <επώνυμο> >τελικός βαθμός>

.....

ΜΕΣΟΣ ΟΡΟΣ ΤΑΞΗΣ

<μέσος όρος>

Κάθε μία από τις παραπάνω λειτουργίες 1-5 θα πρέπει να υλοποιείται σαν μία ξεχωριστή συνάρτηση. Η βάση θα πρέπει να υλοποιηθεί σαν ένα διάνυσμα 30 δεικτών σε δομές στοιχείων τύπου `StudentType`. Το απαιτούμενο πλήθος δομών στοιχείων τύπου `StudentType` θα δημιουργείται αφού διαβαστεί το πλήθος των μαθητών από το αρχείο `project2-askhsh2.dat`.

Τι θα πρέπει να παραδώσετε

Το αρχείο με τον πηγαίο κώδικά σας με όνομα της μορφής `x_set2_exer2.c`, , όπου `x` θα το αντικαταστήσετε με τον κωδικό της ομάδας σας (π.χ. `LAB101030_set2_exer2.c`). Προσοχή στη στοίχιση του κώδικα και τα σχόλια, τα οποία θα πρέπει να είναι επεξηγηματικά του κώδικα. Στις πρώτες γραμμές του πηγαίου κώδικα να αναφέρετε το όνομά σας και τον αριθμό του φοιτητικού μητρώου σας.