

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

Η γλώσσα προγραμματισμού C

*ΕΡΓΑΣΤΗΡΙΟ 3: Πίνακες, βρόχοι, συναρτήσεις, αρχεία*

10 Μαΐου 2018

Το σημερινό εργαστήριο έχει ως θέμα τους μονοδιάστατους αλλά και τους πολυδιάστατους πίνακες της γλώσσας C. Η C δεν υποστηρίζει άμεσα κάποια δομή πολυδιάστατων πινάκων, αλλά έμμεσα, επιτρέποντας τα στοιχεία των μονοδιάστατων πινάκων να είναι άλλοι μονοδιάστατοι πίνακες, κοκ. Θα χρησιμοποιήσετε φωλιασμένους βρόχους για την επεξεργασία τέτοιων πινάκων, ενώ θα δείτε για πρώτη φορά συναρτήσεις και αρχεία.

Τις παρακάτω ασκήσεις θα τις ολοκληρώσετε κατά προτίμηση μέχρι τη μεθεπόμενη εβδομάδα. Η παράδοσή τους θα γίνει όπως έγινε με τα προηγούμενα εργαστήρια. Υπενθυμίζεται η στοίχιση και ο σχολιασμός που πρέπει να έχουν τα προγράμματά σας για να είναι κατανοητά.

*Άσκηση 1: «Υπολογισμός μήκους καμπύλης στο επίπεδο»*

Εκκινήστε το πρόγραμμα *Code::Blocks*. Δημιουργήστε μια καινούργια εργασία και ονομάστε την *lab3*. Δημιουργήστε επίσης ένα νέο αρχείο προγράμματος, με όνομα *ONOMA\_lab3.1.c*, όπου αντί για *ONOMA* θα βάλετε το ονοματεπώνυμό σας. Συνδέστε το με την εργασία *lab3*.

Στην άσκηση αυτή θα γράψετε ένα πρόγραμμα, το οποίο θα υπολογίζει το μήκος μιας καμπύλης στο επίπεδο. Η καμπύλη θα δίνεται ως συνάρτηση, είτε σε καρτεσιανές είτε σε πολικές συντεταγμένες. Σε καρτεσιανές συντεταγμένες, μια καμπύλη εκφράζεται από συνάρτηση της μορφής  $y = f(x)$ , ενώ σε πολικές συντεταγμένες, μια καμπύλη εκφράζεται από συνάρτηση της μορφής  $r = f(\theta)$ . Η συνάρτηση θα δίνεται μέσω ενός αρχείου, το οποίο θα περιέχει τιμές της συνάρτησης για πολύ κοντινές τιμές του  $x$  ή του  $\theta$ .

Το αρχείο αυτό θα έχει την ακόλουθη μορφή:

- Πρώτο στοιχείο είναι το πλήθος  $n$  των σημείων της καμπύλης.
- Δεύτερο στοιχείο είναι η κατώτερη τιμή της ανεξάρτητης μεταβλητής  $x$  ή  $\theta$ .
- Τρίτο στοιχείο είναι το βήμα αύξησης της ανεξάρτητης μεταβλητής.
- Ακολουθούν τόσες τιμές της συνάρτησης  $f$ , όσα και τα σημεία της καμπύλης.

Θεωρήστε ότι όλες οι τιμές εκτός του  $n$  είναι τύπου *double*.

Όσο αφορά τους υπολογισμούς που πρέπει να υλοποιήσετε στον κώδικά σας, από την ελληνική Wikipedia βρίσκουμε (σε περίληψη):

Το μήκος  $L_{AB}$  μίας καμπύλης μεταξύ δύο σημείων A και B της καμπύλης μπορεί να υπολογισθεί μέσω του ολοκληρώματος:

$$L_{AB} = \int_A^B ds$$

όπου  $ds$  είναι το στοιχειώδες μήκος της καμπύλης, το οποίο εκφράζεται βάσει του Πυθαγορείου Θεωρήματος.

Σε καρτεσιανές συντεταγμένες, το στοιχειώδες μήκος μίας καμπύλης εκφράζεται ως:

$$ds^2 = dx^2 + dy^2$$

όπου  $dx$  μία στοιχειώδης μετατόπιση κατά τον άξονα  $x$  και  $dy$  μία στοιχειώδης μετατόπιση κατά τον άξονα  $y$ . Αν η καμπύλη δίνεται υπό τη μορφή μίας συνάρτησης  $y=f(x)$ , το μήκος  $L_{AB}$  μπορεί να υπολογισθεί ως εξής:

$$L_{AB} = \int_{x_A}^{x_B} \sqrt{1 + \left(\frac{dy(x)}{dx}\right)^2} dx$$

όπου  $x_A$  και  $x_B$  οι τιμές στον άξονα  $x$  που αντιστοιχούν στα σημεία A και B.

Σε πολικές συντεταγμένες το στοιχειώδες μήκος της καμπύλης εκφράζεται ως:

$$ds^2 = dr^2 + r^2 d\theta^2$$

Συνεπώς, αφού κάθε καμπύλη περιγράφεται από μια συνάρτηση  $r=f(\theta)$  που μας δίνει την απόσταση  $r$  κάθε σημείου της καμπύλης από την αρχή των αξόνων για δεδομένη τιμή της γωνίας  $\theta$ , μπορούμε να χρησιμοποιήσουμε τη γωνία  $\theta$  ως παράμετρο στο ολοκλήρωμα που μας δίνει το μήκος  $L_{AB}$ . Δηλαδή:

$$L_{AB} = \int_A^B \sqrt{\left(\frac{dr(\theta)}{d\theta}\right)^2 + [r(\theta)]^2} d\theta$$

Επομένως, σύμφωνα με τα παραπάνω, για να υπολογίσετε το μήκος μιας καμπύλης μεταξύ δύο σημείων A και B, θα πρέπει να υπολογίσετε την παράγωγο της συνάρτησης της καμπύλης και στη συνέχεια το ολοκλήρωμα της αντίστοιχης έκφρασης.

Δεδομένου ότι η συνάρτηση δίνεται προσεγγιστικά ως ακολουθία πολύ κοντινών σημείων, οι παράγωγοι θα υπολογίζονται ως πηλίκα, και τα ολοκληρώματα ως αθροίσματα:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, \quad \int_{x_0=a}^{x_n=b} g(x) dx = \sum_{i=0}^{n-1} \{g(x_i) * (x_{i+1} - x_i)\}$$

Γράψτε κατ' αρχήν μια συνάρτηση, η οποία να υπολογίζει την παράγωγο της  $f$  για κάθε σημείο της καμπύλης, με βάση το πιο πάνω πηλίκο. Η συνάρτηση αυτή θα δέχεται ως παραμέτρους το πλήθος των σημείων, την αρχική τιμή της ανεξάρτητης μεταβλητής, το βήμα αύξησης της ανεξάρτητης μεταβλητής, και δύο πίνακες, από τους οποίους ο πρώτος θα περιέχει τις τιμές της  $f$  και ο δεύτερος θα περιέχει τις τιμές της παραγώγου. Η συνάρτηση δεν επιστρέφει κάποιο αποτέλεσμα, παρά μόνο συμπληρώνει το δεύτερο από τους δύο πίνακες.

Γράψτε δύο ακόμα συναρτήσεις, οι οποίες να υπολογίζουν το κατάλληλο ολοκλήρωμα για καθένα από τα δύο συστήματα συντεταγμένων που υποστηρίζονται, με βάση το πιο πάνω άθροισμα. Οι συναρτήσεις αυτές θα δέχονται τους ίδιους τύπους παραμέ-

τρων με τη συνάρτηση υπολογισμού της παραγώγου. Μάλιστα, θα πρέπει να καλούν τη συνάρτηση αυτή, ώστε να συμπληρώνεται ο δεύτερος πίνακας. Για να εισάγετε τον υπολογισμό της τετραγωνικής ρίζας, χρησιμοποιήστε τη συνάρτηση *sqrt()* τύπου *double* από τη βιβλιοθήκη *math*. Οι δύο συναρτήσεις θα επιστρέφουν αποτέλεσμα τύπου *double*.

Ξεκινήστε τώρα τη *main()* με την ανάγνωση του αρχείου εισόδου. Η βασική δομή αρχείου στη C ονομάζεται *FILE* (με κεφαλαία!), την οποία προς το παρόν θα τη χρησιμοποιείτε απλά με το όνομά της, συνήθως ως διεύθυνση με τη βοήθεια του τελεστή '\*', όπως εξηγείται στη συνέχεια. Θα χρησιμοποιήσετε τις εξής συναρτήσεις από τη βιβλιοθήκη *stdio*:

- *fopen()* : Με δύο παραμέτρους, (α) τη συμβολοσειρά με το όνομα του αρχείου και (β) τη συμβολοσειρά του τρόπου προσπέλασης ("r" για ανάγνωση ή "w" για εγγραφή), ανοίγει το αρχείο και επιστρέφει δείκτη σε αυτό ή 0 αν παρουσιαστεί κάποιο σφάλμα.
- *fscanf()* : Με πρόσθετη πρώτη παράμετρο το δείκτη στο αρχείο, οι υπόλοιπες παράμετροι είναι ταυτόσημες με εκείνες της *scanf()*.
- *fclose()* : Με μία παράμετρο, το δείκτη στο αρχείο, κλείνει το αρχείο.

Η συνάρτηση *fclose()* δεν είναι απαραίτητο να καλείται στο τέλος ενός προγράμματος, μια που ο τερματισμός του προγράμματος κλείνει όλα τα ανοικτά αρχεία.

Για τη χρήση των παραπάνω, θα πρέπει να δηλώσετε τοπικά στη *main()* το δείκτη στο αρχείο που θέλετε να ανοίξετε, έστω *f*:

```
FILE * f;
```

Στη συνέχεια θα πρέπει να ζητήσετε από το χρήστη να σας δώσει το όνομα του αρχείου που θέλει να ανοίξετε. Δηλώστε έναν πίνακα χαρακτήρων, έστω *fname* και έστω μέγιστου μεγέθους 100, και χρησιμοποιήστε την *gets()* ή την *scanf()* για να διαβάσετε το όνομα του αρχείου. Τώρα μπορείτε να καλέσετε την *fopen()*, ως εξής:

```
f = fopen(fname, "r");
```

Για ευελιξία σε πιθανά σφάλματα, υλοποιήστε ένα μικρό βρόχο, με τον οποίο επαναληπτικά θα ζητάτε ένα όνομα αρχείου, όσο η συνάρτηση επιστρέφει 0.

Έχοντας υλοποιήσει τον κώδικα που ανοίγει το αρχείο με όνομα που περιέχει η συμβολοσειρά *fname*, προσθέστε τις δηλώσεις των μεταβλητών της άσκησης, και χρησιμοποιήστε την *fscanf()* για να διαβάσετε τους τρεις πρώτους αριθμούς και επαναληπτικά τις τιμές της συνάρτησης.

Για να προχωρήσετε, θα πρέπει να μάθετε το σύστημα συντεταγμένων στο οποίο ορίζεται η συνάρτηση της καμπύλης. Γράψτε τον κώδικα που ζητάει από το χρήστη την πληροφορία αυτή.

Υλοποιήστε τώρα τον κώδικα που καλεί την κατάλληλη συνάρτηση υπολογισμού ολοκληρώματος, και στη συνέχεια εκτυπώνει στην οθόνη το αποτέλεσμα του υπολογισμού.

Μεταφράστε και δοκιμάστε το πρόγραμμά σας με τη βοήθεια κατάλληλου αρχείου εισόδου που μπορείτε να δημιουργήσετε με οποιοδήποτε επεξεργαστή απλού κειμένου.

Αποθηκεύστε και υποβάλετε το αρχείο του προγράμματός σας, και συνεχίστε με την επόμενη άσκηση.

*Άσκηση 2: «Σχεδίαση γεωμετρικών σχημάτων σε δισδιάστατο πίνακα»*

Δημιουργήστε ένα καινούργιο αρχείο προγράμματος και ονομάστε το *ONOMA\_lab3.2.c*, με το ονοματεπώνυμό σας ως *ONOMA*. Συνδέστε το με την εργασία *lab3* και αποσυνδέστε από αυτήν το αρχείο προγράμματος της προηγούμενης άσκησης.

Στην άσκηση αυτή θα γράψετε ένα πρόγραμμα σχεδίασης απλών γεωμετρικών σχημάτων σε δισδιάστατο πίνακα. Πιο συγκεκριμένα, θα σχεδιάσετε τρίγωνα, τετράγωνα, παραλληλόγραμμα, τραπέζια και κύκλους, διαφόρων μεγεθών. Κάθε σχήμα θα πρέπει να εκτυπώνεται μέσα από κατάλληλη συνάρτηση, η οποία θα δέχεται ως παραμέτρους τα απαραίτητα στοιχεία που απαιτούνται για κάθε σχήμα, και θα επιστρέφει 0 για επιτυχημένη και 1 για αποτυχημένη εκτύπωση. Η εκτύπωση θα γίνεται με κάποιον χαρακτήρα – πχ. '\*'.

Δηλώστε ένα δισδιάστατο πίνακα χαρακτήρων μεγέθους 60×60. Ξεκινήστε το πρόγραμμα με ερώτηση προς το χρήστη για το σχήμα που θέλει να εκτυπωθεί. Αφού διαβαστεί η απάντηση, θα πρέπει κατάλληλη εντολή *switch* να διαχειρίζεται κάθε σχήμα αναλυτικότερα, και στη συνέχεια να το εκτυπώνει. Όλα τα μεγέθη των σχημάτων θα πρέπει να είναι σε πλήθος χαρακτήρων που θα εκτυπωθούν.

Για τρίγωνο – όπου η μία ακμή λαμβάνεται ως βάση και τοποθετείται οριζόντια στον πίνακα, θα πρέπει να ζητείται εύρος βάσης, ύψος, καθώς και η σχετική απόσταση της επάνω κορυφής από την κάθετο στη βάση στην κάτω αριστερή κορυφή του τριγώνου. Για παράδειγμα, ένα όρθιο ισοσκελές τρίγωνο εύρους βάσης 10 και οποιουδήποτε ύψους, θα έχει σχετική απόσταση 5 χαρακτήρων, ενώ ένα ορθογώνιο τρίγωνο με την ορθή γωνία κάτω αριστερά και οποιουδήποτε μεγέθους θα έχει σχετική απόσταση 0 χαρακτήρων. Αν η επάνω κορυφή είναι αριστερότερα από την κάτω αριστερή κορυφή, τότε η σχετική απόσταση θα είναι αρνητική.

Για τετράγωνο – όπου πάλι η μία ακμή λαμβάνεται ως βάση και τοποθετείται οριζόντια στον πίνακα, θα πρέπει να ζητείται απλά το εύρος βάσης.

Για παραλληλόγραμμο – όπου και πάλι η μία ακμή λαμβάνεται ως βάση και τοποθετείται οριζόντια στον πίνακα, θα πρέπει να ζητείται το εύρος βάσης, το ύψος, και η σχετική απόσταση της επάνω αριστερής κορυφής από την κάθετο στη βάση στην κάτω αριστερή κορυφή του παραλληλογράμμου. Για παράδειγμα ένα ορθογώνιο παραλληλόγραμμο οποιουδήποτε μεγέθους θα έχει σχετική απόσταση 0 χαρακτήρων.

Για τραπέζιο – όπου η μία από τις δύο παράλληλες ακμές λαμβάνεται ως βάση και τοποθετείται οριζόντια στον πίνακα, θα πρέπει να ζητείται το εύρος βάσης, το ύψος και η σχετική απόσταση της επάνω αριστερής και της επάνω δεξιάς κορυφής από την κάθετο στη βάση στην κάτω αριστερή κορυφή του τραπεζίου.

Για κύκλο, θα πρέπει να ζητείται απλά η ακτίνα.

Σε κάθε περίπτωση, θα καλείται η κατάλληλη συνάρτηση για την εκτύπωση του σχήματος, με παραμέτρους τα πιο πάνω στοιχεία. Η συνάρτηση θα πρέπει αρχικά να βρίσκει κάποιο σημείο κοντά στο κέντρο του πίνακα όπου θα γίνει η εκτύπωση, έτσι ώστε το σχήμα να μη βγει εκτός του πίνακα. Αν το σχήμα δε χωράει στον πίνακα, η συνάρτηση επιστρέφει 1. Στη συνέχεια, η συνάρτηση θα πρέπει να βρίσκει τα σημεία του πίνακα που θα πρέπει να σημειωθούν με το χαρακτήρα σχεδίασης. Θεωρήστε τον πίνακα σα σύστημα συντεταγμένων, και για όλες τις περιπτώσεις εκτός του κύκλου, βρείτε τις συντεταγμένες των κορυφών του σχήματος, και με βάση τις εξισώσεις των ακμών, βρείτε εκείνα τα σημεία του πίνακα που πέφτουν πιο κοντά στις ακμές, προ-

χωρώντας επαναληπτικά από κάθε κορυφή μέχρι την επόμενη. Ειδικά για κύκλο, ξεκινήστε με τις συντεταγμένες του κέντρου, και με βάση την εξίσωση του κύκλου, βρείτε τα σημεία του πίνακα που είναι πιο κοντά στην καμπύλη.

Για τις ακμές μπορείτε να χρησιμοποιήσετε την ακόλουθη εξίσωση ευθείας που περνάει από δύο σημεία  $(x1, y1)$  και  $(x2, y2)$ :

$$\frac{y - y1}{x - x1} = \frac{y2 - y1}{x2 - x1}$$

και για τον κύκλο την εξίσωση κύκλου με κέντρο το σημείο  $(x0, y0)$  και ακτίνα  $r$ :

$$(x - x0)^2 + (y - y0)^2 = r^2$$

Σε όποια από τα σχήματα μπορείτε, να εκμεταλλευτείτε τη συμμετρία που έχουν, ώστε να ελαχιστοποιήσετε τους υπολογισμούς.

Για κάθε σχήμα που θα ολοκληρώνετε, εκτυπώστε το αποτέλεσμα στην οθόνη, καθώς και σε ένα αρχείο με κατάλληλο όνομα, ανοίγοντάς το με τη συνάρτηση *fopen()*, με δεύτερη παράμετρο "w" για εγγραφή, γράφοντας σε αυτό με τη συνάρτηση *fprintf()*, η οποία λειτουργεί όπως η *printf()*, με μια πρόσθετη πρώτη παράμετρο με το δείκτη στο αρχείο, και στο τέλος κλείνοντάς το με τη συνάρτηση *fclose()*.

Μεταφράστε και δοκιμάστε τον κώδικά σας για κάθε σχήμα του οποίου τον κώδικα ολοκληρώνετε.

Όταν γράψετε τον κώδικα για όλα τα σχήματα, αποθηκεύστε και υποβάλετε το αρχείο του προγράμματος.