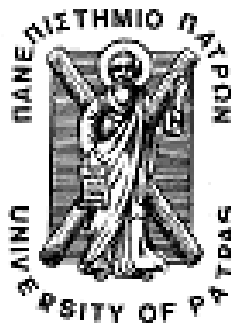


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΚΗΣ



3^ο ΣΕΤ ΑΣΚΗΣΕΩΝ

Οι ασκήσεις αυτού του φυλλαδίου καλύπτουν τα παρακάτω θέματα:

- Δείκτες
- Δομές
- Αρχεία

Το τρίτο σύνολο ασκήσεων θα πρέπει να πραγματοποιηθεί τις εβδομάδες 19.12.2016-15.01.2017 (για κάθε υποερώτημα ο φοιτητής θα πρέπει να έχει το αντίστοιχο αρχείο κώδικα C αποθηκευμένο στο λογαριασμό του στο diogenis.ceid.upatras.gr το οποίο θα πρέπει να έχει μεταγλωττιστεί χωρίς σφάλματα, ενώ για το σύνολο των ερωτημάτων καλό είναι να ετοιμάζει και ένα αρχείο κειμένου με τις απαντήσεις του).

Σημείωση: αν χρειάζεται συμπερίληψη της βιβλιοθήκης <math.h>, τότε θα πρέπει η μεταγλώττιση να πραγματοποιηθεί με την οδηγία -lm, δηλαδή η μεταγλώττιση να γίνει ως εξής:

gcc -lm ονομα_αρχείου

Άσκηση 1

Να ορίσετε μία δομή που θα αποθηκεύει τα στοιχεία ενός τραπεζικού λογαριασμού με πεδία: όνομα καταθέτη, ποσό κατάθεσης και ημερομηνία κατάθεσης. Γράψτε πρόγραμμα το οποίο: (i) θα δηλώνει ένα πίνακα με το όνομα *accounts* που θα αποθηκεύει ένα σύνολο n τραπεζικών λογαριασμών, (ii) θα διαβάζει τα στοιχεία των n λογαριασμών από τον χρήστη, (iii) θα διαβάζει το όνομα ενός καταθέτη και θα εκτυπώνει όλες τις πληροφορίες για όλους τους λογαριασμούς με το συγκεκριμένο όνομα. Αν οι καταθέσεις είναι περισσότερες από μία θα εκτυπώνει επίσης το άθροισμα όλων των ποσών κατάθεσης.

Τα ζητούμενα (ii) και (iii) να γίνουν με χρήση κατάλληλων συναρτήσεων και η υλοποίηση του πίνακα να γίνει με δείκτες.

Άσκηση 2

Ένας καθηγητής θέλει να καταχωρεί τα στοιχεία των φοιτητών (N σε σύνολο) που έχουν επιλέξει το μάθημα του και στην συνέχεια να εμφανίζεται στην οθόνη του το ονοματεπώνυμο των φοιτητών που έχουν γράψει τον μεγαλύτερο βαθμό και ποιος είναι αυτός ο βαθμός, το ονοματεπώνυμο των φοιτητών που έχουν δώσει περισσότερες φορές το μάθημα του και ποιος είναι αυτός ο αριθμός καθώς και το ονοματεπώνυμο των φοιτητών που είναι μεγαλύτερο έτος και ποιο έτος είναι αυτό .

Στο πρόγραμμα να χρησιμοποιείται δομή με τα παρακάτω στοιχεία φοιτητών:

Όνομα φοιτητή

Επώνυμο φοιτητή

Βαθμός μαθήματος

Φορές που έδωσαν το μάθημα

Έτος φοιτητή

Να δώσετε δύο υλοποιήσεις μία με πίνακα και μία με δείκτη.

Άσκηση 3

Θεωρήστε δύο συμβολοσειρές A και B καθεμία μέγιστου μήκους $M-1$. Κάθε συμβολοσειρά θεωρούμε ότι μοντελοποιεί ένα κείμενο συνεπώς αποτελείται από μία ακολουθία λέξεων με κάθε λέξη να αποτελείται από συνεχόμενες εμφανίσεις αριθμητικών και αλφαβητικών χαρακτήρων, όλων του Αγγλικού αλφαβήτου. Υποθέτουμε ότι οι λέξεις οριοθετούνται με ένα ή περισσότερα κενά και ότι γίνεται διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων.

Να υλοποιήσετε κώδικα που διαβάζει το περιεχόμενο των A και B και για κάθε λέξη της A δημιουργεί μία ακολουθία με τις θέσεις εμφάνισής της στη B (ως θέση εμφάνισης μία λέξης εννοούμε τον αριθμό του κελιού του B στο οποίο η λέξη αρχίζει να εμφανίζεται). Η υλοποίηση της ακολουθίας γίνεται είτε με λίστα, είτε με πίνακα.

Στη συνέχεια ο κώδικας για κάθε διακριτή λέξη του A , εκτυπώνει τις θέσεις εμφάνισης της στο B .

Υπόδειξη: Για την εύρεση των λέξεων που εμφανίζονται σε μία συμβολοσειρά προτείνεται να χρησιμοποιηθεί η συνάρτηση `char * strtok(char *s, char * ct)` της βιβλιοθήκης `<string.h>`. Για την σύγκριση αλφαριθμητικών προτείνεται η συνάρτηση `int strcmp(cs, ct)` της βιβλιοθήκης `<string.h>`.

Άσκηση 4

Καλείστε να υλοποιήσετε ένα απλουστευμένο παιχνίδι στρατηγικής. Παίζεται από 2 παίκτες οι οποίοι αγοράζουν μονάδες για τον στρατό τους. Οι αγορές γίνονται εκ περιτροπής και οι μονάδες έχουν διαφορετικό κόστος. Νικητής είναι αυτός που θα συγκεντρώσει πολύ μεγαλύτερο στρατό από τον αντίπαλό του ή συγκεντρώσει αρκετά χρήματα. Πιο συγκεκριμένα υπάρχουν μονάδες (εργάτες) που προσφέρουν λεφτά και μονάδες στρατιωτικές οι οποίες καταναλώνουν χρήματα. Ο κάθε παίχτης διαλέγει τι θα αγοράσει στον γύρο του ανάλογα με την στρατηγική του.

Υλοποίηση: Θα δημιουργήσετε μία δομή (struct) με όνομα units και θα περιέχει όλα τα χαρακτηριστικά της μονάδας. Στη συνέχεια θα δημιουργήσετε 2 πίνακες από units 1 για κάθε παίκτη. Θα κάνετε τις κατάλληλες αρχικοποιήσεις (εδώ μπορείτε να δίνετε πληροφορίες για το κόστος των μονάδων όνομα και άλλα από αρχείο). Στη συνέχεια ένας βρόχος επανάληψης θα δέχεται τις επιλογές του κάθε παίκτη και θα υπολογίζει τα χρήματα και τον στρατό του καθενός. Εάν υπάρχει νικητής τότε θα εκτυπώνεται και το πρόγραμμα θα τερματίζει.

Π.χ. δομής :

```
struct unit {
    char name[20]; //Το όνομα της μονάδας
    int att;        // Η ισχύς της μονάδας.
    int cost;      // Το κόστος για την αγορά της μονάδας
    int sup;       // Το κόστος συντήρησης της μονάδας ανά γύρο.
    int code;      // βοηθητικό πεδίο.
    int num;       // Αριθμός συγκεκριμένων μονάδων που διαθέτει ο παίχτης.
};
```

Παραδείγματα μονάδων.

Όνομα	επίθεση	κόστος	συντήρηση
Peon	0	20	-15
Grunt	2	30	5
Troll	3	20	10

...

Άσκηση 5

Να υλοποιήσετε ένα απλό diff/patch πρόγραμμα το οποίο θα έχει τη δυνατότητα να συγκρίνει δύο αρχεία εισόδου και να καταγράφει σε ένα νέο αρχείο τις γραμμές στις οποίες διαφέρουν. Επίσης θα έχει και μια δεύτερη λειτουργία όπου θα δέχεται ένα αρχείο εισόδου και ένα αρχείο διαφορών και θα παράγει το διαφοροποιημένο αρχείο. Η επιλογή ποια από τις δύο λειτουργίες θα εκτελεστεί θα γινετια με επιλογή του χρήστη.

file_old.txt	file_new.txt	diff.txt
--------------	--------------	----------

<p>Hello, How are you? I hope all is well. br, Joe</p>	<p>Hello, How do you do? I hope all is well. Best regards, Joe</p>	<p>---Line 2 file_old.txt How are you? file_new.txt How do you do? ---Line 4 file_old.txt br, file_new.txt Best regards,</p>
--	--	--