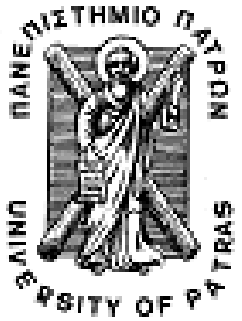
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΑΣΤΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

4^{ο} ΣΕΤ ΑΣΚΗΣΕΩΝ

Οι ασκήσεις αυτού του φυλλαδίου καλύπτουν τα παρακάτω θέματα και δίνονται ενδεικτικά οι αντίστοιχες ενότητες στο βιβλίο The GNU C Programming Tutorial που μπορείτε συμβουλευτείτε (http://crasseux.com/books/ctutorial/) .

- Δείκτες
- Δομές
- Αρχεία

Το τέταρτο σύνολο ασκήσεων θα εξετασθεί μαζί με το τρίτο την εβδομάδα 07-13/01/13 (ο κάθε φοιτητής θα εξεταστεί την ώρα του τμήματος του, 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 θα πρέπει να έχει φέρει και μια εκτυπωμένη αναφορά με τις απαντήσεις του).
Η τελευταία άσκηση του φυλλαδίου είναι προαιρετική, χωρίς βαθμολογική συνεισφορά, για όσους φοιτητές θέλουν και έχουν χρόνο να ασχοληθούν.

Άσκηση 1η

Γράψτε ένα πρόγραμμα που θα ζητάει από το χρήστη έναν αριθμό N και στη συνέχεια θα δημιουργεί μία απλά διασυνδεδεμένη λίστα ακεραίων αριθμών L1 με αριθμούς τους οποίους θα ζητάει από το χρήστη. Επίσης θα δημιουργεί και μια άλλη διασυνδεδεμένη λίστα ίσου μήκους L2, της οποίας κάθε στοιχείο προκύπτει με βάση την συνάρτηση $f(x)=x^3$, σε σχέση με το αντίστοιχο στοιχείο της πρώτης λίστας. Π.χ. για L1 = {1 , 4, 6, 10} και $f(x)=x^3$, τότε L2 = {1 , 64, 216, 1000}. Στο τέλος το πρόγραμμα θα τυπώνει τις λίστες L1 και L2.

Παράδειγμα εκτέλεσης:

```
How many elements in the list? 4
Element 1: 1
Element 2: 4
Element 3: 6
Element 4: 10
List L1: 1 4 6 10
List L2: 1 64 216 1000
```

Άσκηση 2η

Γράψτε πρόγραμμα που μετράει τον αριθμό των bytes που περιέχει ένα αρχείο. Το όνομα του αρχείου θα δίνεται από τον χρήστη.

Άσκηση 3η

Γράψτε μία συνάρτηση **replace(input, initial, replacement)** η οποία παίρνει ως παράμετρο τα αλφαριθμητικά **input**, **initial** και **replacement** και επιστρέφει ένα νέο αλφαριθμητικό στο οποίο όλες οι εμφανίσεις του **initial** στο **input** έχουν αντικατασταθεί με το **replacement**. Η συνάρτηση θα πρέπει να δουλεύει με αλφαριθμητικά οποιουδήποτε μήκους (επομένως χρησιμοποιήστε δείκτες). Το κυρίως πρόγραμμα θα ζητάει από το χρήστη τρία αλφαριθμητικά μεγέθους το πολύ N (το N θα το δίνει επίσης ο χρήστης) και θα τυπώνει το αποτέλεσμα της replace.

Π.χ. (με έντονα η είσοδος του χρήστη):

```
Dwse megisto mege8os string: 30
Dwse string eisodou (mege8os to poly 30):
123456789012345678901234
Dwse string pros antikatastasi(mege8os to poly 30): 456
Dwse neo string (mege8os to poly 30): TesseraPenteEksi
123TesseraPenteEksi7890123TesseraPenteEksi78901234
```

Άσκηση 4η

Σας ζητείται να υλοποιήσετε μία απλή εκδοχή του επιτραπέζιου παιχνιδιού Mastermind (http://en.wikipedia.org/wiki/Mastermind_(board_game)), χωρίς γραφικό περιβάλλον. Τα διαθέσιμα χρώματα είναι έξι (κόκκινο, μπλε, πράσινο, κίτρινο, μωβ, άσπρο), και αναπαρίστανται από τα κεφαλαία λατινικά γράμματα R,B,G,Y,P,W. Το πρόγραμμα αρχικά ζητά από τον χρήστη το μήκος του κρυφού κωδικού (επιτρέπεται από 1 έως 100) και παράγει μία τυχαία ακολουθία των παραπάνω χρωμάτων (πιθανά με επαναλαμβανόμενα χρώματα) η οποία είναι ο κρυφός κωδικός. Στη συνέχεια αρχίζει η διαδικασία όπου ο χρήστης δίνει ακολουθίες χρωμάτων με το καθορισμένο μήκος και το πρόγραμμα του απαντά με 2 πληροφορίες: Τον αριθμό των χρωμάτων που βρήκε σωστά όσον αφορά θέση και χρώμα, και τον αριθμό των χρωμάτων που βρήκε σωστά αλλά σε λάθος θέση. Αν ο χρήστης βρει όλα τα χρώματα στη σωστή θέση, το πρόγραμμα του ανακοινώνει ότι νίκησε (και πόσες προσπάθειες χρειάστηκε). Αν ο χρήστης αντί για την ακολουθία χρωμάτων πληκτρολογήσει SURRENDER το πρόγραμμα του ανακοινώνει ότι έχασε και φανερώνει τον κωδικό. Αν σε κάποια ακολουθία ο χρήστης πληκτρολογήσει μη έγκυρο γράμμα το πρόγραμμα αγνοεί την συγκεκριμένη ακολουθία.

Παράδειγμα εκτέλεσης:

```
Please define secret code length: 6
Try 1: RRRWWB
  Color and position correct: 1
  Only color correct:         2
Try 2: RBBBBH
```

```
Error sequence, please try again.
Try 2: RBBBBB
  Color and position correct: 3
  Only color correct:         0
Try 3: RBBRBB
  Color and position correct: 4
  Only color correct:         0
Try 4: RBGRBGGG
Code sequence has to be exactly 6 colors
Try 4: RBGRBG
  Color and position correct: 6
  Only color correct:         0
Congratulations! You have won the game after 4 tries.
```

Παράδειγμα εκτέλεσης όπου ο χρήστης χάνει:

```
Please define secret code length: 6
Try 1: BWWGGG
  Color and position correct: 2
  Only color correct:         2
Try 2: SURRENDER
You have lost the game. The secret code was GWBYGB.
```

Παραγωγή τυχαίων χρωμάτων:

Για την παραγωγή τυχαίων αριθμών (χρωμάτων) χρησιμοποιήστε την γεννήτρια τυχαίων αριθμών rand() της γλώσσας C όπως στο παρακάτω παράδειγμα.

```
#include <stdlib.h>
#include <time.h>

// συνάρτηση για την παραγωγή τυχαίου αριθμού μέσα σε
// καθορισμένο πεδίο τιμών [low, high]
int RandomInteger(int low, int high) {
  int k;
  double d;
  d = (double) rand() / ((double)RAND_MAX + 1);
  k = (int) (d * (high - low + 1));
  return (low + k);
}

int main() {
  int i, number;

  // αρχικοποίηση γεννήτριας παραγωγής τυχαίων αριθμών
  srand((int)time(NULL));

  // κλήση της RandomInteger για την παραγωγή 10 τυχαίων
  // αριθμών στο πεδίο τιμών [1,9]
  for (i = 0; i < 10; i++) {
    number = RandomInteger(1, 9);
    printf("%d\n", number);
  }
}
```

Άσκηση 5η

Φτιάξε πρόγραμμα το οποίο θα μετατρέπει τα περιεχόμενα ενός αρχείου και θα τα επαναφέρει με την εξής απλή μέθοδο. Το κλειδί θα είναι ένα string το οποίο θα δίνεται από το χρήστη και από το οποίο το πρόγραμμα θα δημιουργεί ένα άθροισμα Α προσθέτοντας τους κωδικούς ASCII των χαρακτήρων του. Ο χρήστης θα δίνει επίσης ένα αρχείο, σε κάθε χαρακτήρα του οποίου το πρόγραμμα θα προσθέτει το άθροισμα Α δημιουργώντας έτσι νέους χαρακτήρες. Η επαναφορά του αρχείου γίνεται με αφαίρεση του αθροίσματος Α για να προκύψουν οι αρχικοί χαρακτήρες. Το πρόγραμμα θα παρουσιάζει ένα μενού στο οποίο ο χρήστης θα επιλέγει αν θέλει να κάνει μετατροπή ή επαναφορά ενός αρχείου χρησιμοποιώντας το string που θα δίνει.

Άσκηση 6η

Φτιάξτε ένα απλό παιχνίδι "κρεμάλας" ως εξής: Σε ένα αρχείο με όνομα "words.txt" καταγράφονται αγγλικές λέξεις, μία σε κάθε γραμμή του αρχείου. Αυτό το αρχείο το φορτώνει το πρόγραμμα και το αποθηκεύει σε ένα πίνακα δομών χρήσιμων για το παιχνίδι. Η συγκεκριμένη δομή κουβαλά την λέξη, τον αριθμό των φανερών (open) γραμμάτων, και ένα δυναμικό πίνακα που περιέχει τις φανερές θέσεις πάνω στη λέξη. Όλα τα άλλα γράμματα πρέπει να παρουσιάζονται ως κρυμμένα (σύμβολο _). Η δομή μπορεί να είναι ως εξής:

```
struct game_word{
  char *word;
  int open_letters_count;
  int *open_letters;
};
```

Η αρχικοποίηση του πίνακα πρέπει να επιβάλει το άνοιγμα 3 τυχαίων θέσεων για κάθε μια εγγραφή του. Στη συνέχεια ο χρήστης επιλέγει είτε να παίξει με μια τυχαία επιλεγμένη λέξη ή να οδηγηθεί στον τερματισμό του παιχνιδιού. Κατα το παιχνίδι ο χρήστης δίνει ένα-ένα τα γράμματα ώστε να συμπληρώσει τη λέξη. Σε κάθε τέτοια είσοδο η σωστή εισαγωγή εμφανίζει την ανανεωμένη λέξη, με το λάθος γράμμα να μην τιμωρείται, δηλαδή να μην υπάρχει κάποιος μετρητής λάθος προσπαθειών.

Τέλος με τη σωστή συμπλήρωση μιας λέξης , αυτή αφαιρείται από την λίστα και το παιχνίδι συνεχίζεται.

Παράδειγμα όπου στο πρόγραμμα έχει φορτωθεί αρχείο 12 λέξεων που περιέχει μεταξύ άλλων λέξεων το "standard":

```
Loaded words : 12
Three letters appear for each word.
Type 1 to play a new word (12 available words).
```

```
Type 2 to exit.
:1
    Word is "__a___rd".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : a
    Word is "__a__ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : b
    Word is "__a__ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : s
    Word is "s_a__ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : t
    Word is "sta__ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : m
    Word is "sta__ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : n
    Word is "stan_ard".
    Type 1 to give letter.
    Type 2 to go back to menu.
    :1
    Letter : d
You found word :"standard". Removing it from list
Type 1 to play a new word (11 available words).
Type 2 to exit.
:2
Exiting.
```

Άσκηση 7η

Σας ζητείτε να αποθηκεύσετε σε έναν πίνακα μεγέθους Ν στοιχεία τύπου struct employee. Κάθε καινούριο στοιχείο αποθηκεύεται στην τελευταία άδεια θέση του πίνακα και μπορεί να διαγραφεί μόνο το τελευταίο στοιχείο που έχει εισαχθεί στον πίνακα. Δίνεται:

```
struct employee {
  char *firstName;
  char *lastName;
  int age;
};
```

Υλοποιήστε τις παρακάτω συναρτήσεις:

```
// δημιουργία του πίνακα
// pos : πρώτη ελεύθερη θέση στον πίνακα
// επιστρέφει πίνακα με Ν στοιχεία τύπου struct employee*
struct employee* create( int *pos);

// εισαγωγή στοιχείου item στον πίνακα pin
// pos : πρώτη ελεύθερη θέση στον πίνακα
void ins(struct employee* pin, int *pos, struct employee* item);

// διαγραφή στοιχείου από τον πίνακα
// pos : πρώτη ελεύθερη θέση στον πίνακα
struct employee* del(struct employee* pin, int *pos);

// εκτύπωση πίνακα
// pos : πρώτη ελεύθερη θέση στον πίνακα
void printpin(struct employee* pin, int *pos);

// ταξινόμηση πίνακα
// pos : πρώτη ελεύθερη θέση στον πίνακα
// field : το πεδίο με βάση το οποίο θέλουμε να γίνει η ταξινόμηση
// direction : αν η ταξινόμηση θα είναι αύξουσα τότε (direction=1)
// Αν θα είναι φθίνουσα (direction=-1) αντίστοιχα.
void sort(struct employee* pin, int *pos, char *field, int
direction);
```

Γράψτε την main() η οποία εμφανίζει μενού επιλογών στο χρήστη:

1. Εισαγωγή στοιχείων υπαλλήλου
2. Διαγραφή στοιχείων υπαλλήλου
3. Εμφάνιση όλων των υπαλλήλων
4. Ταξινόμηση Πίνακα
5. Έξοδος

Ανάλογα με την επιλογή του χρήστη ζητούνται τα απαραίτητα στοιχεία (π.χ. για την Εισαγωγή Στοιχείων ζητούνται από τον χρήστη το όνομα, επώνυμο και ηλικία) και καλείται η κατάλληλη συνάρτηση.

<u>ΠΡΟΑΙΡΕΤΙΚΗ ΑΣΚΗΣΗ</u> (ΧΩΡΙΣ ΒΑΘΜΟΛΟΓΙΚΗ ΣΥΝΕΙΣΦΟΡΑ, για όσους θέλουν και έχουν χρόνο να ασχοληθούν)

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 πρώτη λειτουργία θα εκτελείται με την παράμετρο γραμμής εντολών –diff και η δεύτερη με την –patch.

file_old.txt	file_new.txt	diff.txt
Hello, How are you? I hope all is well. br, Joe	Hello, How do you do? I hope all is well. Best regards, Joe	---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,

Παρακάτω είναι ενδεικτικοί τρόποι χρήσης του προγράμματος με τα παραπάνω αρχεία (θεωρώντας ότι το εκτελέσιμο με το πρόγραμμα ονομάζεται ask5):

```
ask5 -diff file_old.txt file_new.txt
```

(Θα παράγει το αρχείο diff.txt)

```
ask5 -patch file_old.txt diff.txt
```

(Θα παράγει το αρχείο file_new.txt)