lab8: Συναρτήσεις

Προετοιμασία:

1. Πλοηγηθείτε στον κατάλογο lab8 ο οποίος πρέπει να βρίσκεται μέσα στον κατάλογο ce120.
2. Ολοκληρώστε τις ασκήσεις της εβδομάδας. Τα αρχεία που θα γράψετε πρέπει να αποθηκευτούν στον κατάλογο lab8 που δημιουργήσατε προηγούμενα.
    - Θυμίζουμε πως η εντολή για να κάνετε compile ένα αρχείο με όνομα **`lab8a.c`** είναι:

      `gcc -Wall -g lab8a.c -o lab8a`

    Αν δεν υπάρχουν λάθη, θα δημιουργηθεί το εκτελέσιμο αρχείο **`lab8a`** το οποίο μπορείτε να εκτελέσετε γράφοντας **`./lab8a`**

Θέματα εργασίας:

- Συναρτήσεις
- Συμβολοσειρές
- Πίνακες μίας διάστασης
- Χρήση εντολών ελέγχου και επανάληψης
- Αναγνωσιμότητα κώδικα (σχόλια, στοίχιση, περιγραφικά ονόματα μεταβλητών, κτλ.)

Ελεγχος ορθότητας:

Για κάθε άσκηση θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου. Πέρα από τους ελέγχους που θα κάνετε εσείς, συνιστούμε να τρέξετε τα προγράμματά σας και με τις δικές μας εισόδους, να αποθηκεύσετε τις αντίστοιχες εξόδους σας σε αρχεία με χρήση ανακατεύθυνσης και μετά να τις συγκρίνετε με τις δικές μας εξόδους με χρήση της εντολής diff.

Για παράδειγμα, αν σας έχουμε δώσει το αρχείο εισόδου in1.txt και την αντίστοιχη έξοδο std1.txt, τρέξτε

```
./lab8a < in1.txt > out1.txt
```

για να αποθηκεύστε την έξοδό σας στο αρχείο out1.txt, και μετά συγκρίνετε

```
diff out1.txt std1.txt
```

Αν δεν υπάρχουν διαφορές, δε θα εμφανιστεί τίποτα. Αν υπάρχουν, θα εμφανιστούν ανά γραμμή

Ασκηση 1

Ξεκινήστε από το έτοιμο **`lab8a.c`** που σας δίνουμε μέσα στον κατάλογο `lab8`. Διαβάστε και τις σημειώσεις της επόμενης σελίδας!

Καλείστε να γράψετε ένα πρόγραμμα το οποίο θα υπολογίζει τον φυσικό λογάριθμο **`ln(x)`** με δύο τρόπους: **(α)** χρησιμοποιώντας τη συνάρτηση **`log`** από το **`math.h`** και **(β)** υπολογίζοντας το αποτέλεσμα της παρακάτω μαθηματικής σειράς.

$$\ln(x) = 2\sum_{n=1}^{limit} \frac{((x-1)/(x+1))^{2n-1}}{(2n-1)}, \gamma\iota\alpha\, x > 0$$

Γράψτε μια συνάρτηση με όνομα **`my_pow`** η οποία παίρνει ως παραμέτρους έναν double αριθμό που αποτελεί τη βάση, και έναν ακέραιο που αποτελεί τον εκθέτη και επιστρέφει το αποτέλεσμα της ύψωσης σε δύναμη της βάσης στον εκθέτη. Το αποτέλεσμα είναι τύπου double. Δεν επιτρέπεται να χρησιμοποιήσετε συνάρτηση ύψωσης σε δύναμη από τη μαθηματική βιβλιοθήκη.

Γράψτε μια συνάρτηση με όνομα **`series_log_factor`** η οποία παίρνει ως παραμέτρους έναν double αριθμό x και έναν ακέραιο n και υπολογίζει και επιστρέφει το αποτέλεσμα της παρακάτω έκφρασης

$$\frac{((x-1)/(x+1))^{2n-1}}{(2n-1)}, \gamma\iota\alpha\, x > 0$$

Γράψτε μια συνάρτηση με όνομα **`series_log`** η οποία παίρνει ως παραμέτρους έναν double αριθμό x και ένα ακέραιο όριο *limit* και υπολογίζει κι επιστρέφει το `ln(x)` χρησιμοποιώντας τον τύπο που σας δώσαμε και τη συνάρτηση **`series_log_factor`**.

Γράψτε τη συνάρτηση `main` η οποία κάνει τα εξής:

1. (Ετοιμο) Εκτυπώνει το μήνυμα "**`Enter x (>0): `**" και στη συνέχεια διαβάζει ένα double αριθμό. Εάν δοθεί μη θετικός αριθμός επαναλαμβάνει το βήμα
2. (Ετοιμο) Εκτυπώνει το μήνυμα **`Enter limit: `** και στη συνέχεια διαβάζει ένα ακέραιο που αναπαριστά το όριο.
   - Ο ακέραιος πρέπει να είναι μεταξύ **0** (μη συμπεριλαμβανομένου) και **N** (συμπεριλαμβανομένου). Εάν δεν είναι, εκτυπώνει **`Invalid value!`** ακολουθούμενο από χαρακτήρα αλλαγής γραμμής και επαναλαμβάνει το βήμα 2. Για τους υπολογισμούς σας χρησιμοποιήστε **N=10**.
3. Υπολογίζει το φυσικό λογάριθμο του x μέσω της συνάρτησης **`series_log`** και εκτυπώνει χαρακτήρα αλλαγής γραμμής και το μήνυμα "**`series ln(x): Y`**", όπου x ο αριθμός που διάβασε εκτυπωμένος με ακρίβεια 3 δεκαδικών ψηφίων και `Y` ο φυσικός λογάριθμος εκτυπωμένος με ακρίβεια 15 δεκαδικών ψηφίων.
4. Υπολογίζει το φυσικό λογάριθμο του x μέσω της συνάρτησης **`log`** από το math.h και εκτυπώνει το μήνυμα "**`##math ln(x): Y`**", όπου x ο αριθμός που διάβασε εκτυπωμένος με ακρίβεια 3 δεκαδικών ψηφίων και `Y` ο φυσικός λογάριθμος εκτυπωμένος με ακρίβεια 15 δεκαδικών ψηφίων.

Σημειώσεις:

1. Για την ύψωση σε δύναμη θα χρησιμοποιήσετε τη συνάρτηση **`my_pow`** που φτιάξατε.
2. Στο πρόγραμμά σας θα χρειαστεί να κάνετε include το αρχείο math.h
3. Στην εντολή μεταγλώττιση θα χρειαστεί να προσθέσετε το directive ***-lm***, δηλαδή

   `gcc -Wall -g lab8a.c -o lab8a -lm`

4. Το πρόγραμμα σας πρέπει να είναι γραμμένο ώστε να μπορεί να αλλαχθεί εύκολα η τιμή του **N**.
5. Για αν δοκιμάσετε το πρόγραμμα σας χρησιμοποιήστε τιμές για το **x** από 0 (μη συμπεριλαμβανομένου) έως και 6.

Αρχεία εισόδου: **`a_in1.txt, a_in2.txt, a_in3.txt`**

Αντίστοιχα αρχεία εξόδου: **`a_std1.txt a_std2.txt`** και **`a_std3.txt`**

Ασκηση 2

Το πρόγραμμα που θα γράψετε για την άσκηση 2 πρέπει να το αποθηκεύσετε σε αρχείο με όνομα **`lab8b.c`** μέσα στον κατάλογο `lab8`.

Γράψτε ένα πρόγραμμα το οποίο κάνει τα εξής:

1. Εκτυπώνει το μήνυμα **`Enter character:`** , και διαβάζει ένα χαρακτήρα από το πληκτρολόγιο.
2. Εκτυπώνει το μήνυμα **`Enter string:`** και διαβάζει μία συμβολοσειρά από το πληκτρολόγιο. Η συμβολοσειρά θα αποθηκευτεί σε πίνακα μέγιστου μεγέθους **`SIZE`**.
3. Στη συμβολοσειρά του βήματος 2, βρίσκει τον πρώτο χαρακτήρα που αντιπροσωπεύει σημείο στίξης. Μετά από αυτόν τον χαρακτήρα ενθέτει τον χαρακτήρα του βήματος 1 στη συμβολοσειρά.

   Η διαδικασία **ΔΕΝ** επιτρέπεται να γίνει με τη χρήση βοηθητικού πίνακα. Κατά την διαδικασία δεν απαιτούμε να ελέγχετε εάν η νέα συμβολοσειρά βγαίνει εκτός ορίων του πίνακα (θεωρήστε ότι δεν βγαίνει).
4. Αφού προσθέσει τον χαρακτήρα στη συμβολοσειρά εκτυπώνει στην οθόνη χαρακτήρα αλλαγής γραμμής, το μήνυμα: **`Enhanced string:`** και αμέσως μετά τη συμβολοσειρά που προκύπτει ως αποτέλεσμα από το βήμα 3 ακολουθούμενη και πάλι από χαρακτήρα αλλαγής γραμμής.

Προσοχή:

- Η άσκηση πρέπει να υλοποιηθεί με χρήση κατάλληλης συνάρτησης από το ctype.h. Επιπλέον, το πρόγραμμα σας πρέπει να είναι γραμμένο ώστε να μπορούν να αλλαχθούν εύκολα τα μεγέθη των πινάκων στους οποίους αποθηκεύονται οι συμβολοσειρές. Στο τελικό πρόγραμμα που θα στείλετε ορίστε **`SIZE 20`**.
- Πρέπει πρώτα να αποθηκεύσετε την αρχική συμβολοσειρά σε πίνακα ακριβώς όπως έχει δοθεί, και μετά να κάνετε τις αλλαγές που περιγράφει το βήμα 3.

Παραδείγματα:

- Εάν η αρχική συμβολοσειρά είναι **`helloworld`** και ο χαρακτήρας είναι **`$`** η τελική συμβολοσειρά είναι ίδια.
- Εάν η αρχική συμβολοσειρά είναι **`hello!world?`** και ο χαρακτήρας είναι **`$`** η τελική συμβολοσειρά είναι **`hello!$world?`**
- Εάν η αρχική συμβολοσειρά είναι **`!,?`** και ο χαρακτήρας είναι **`$`** η τελική συμβολοσειρά είναι **`!$,?`**

Αρχεία εισόδου: **`b_in1.txt, b_in2.txt, b_in3.txt, b_in4.txt`**

Αντίστοιχα αρχεία εξόδου: **`b_std1.txt b_std2.txt, b_std3.txt και b_std4.txt`**

Αποστολή Προγραμμάτων για σχολιασμό και βαθμολόγηση

- Πλοηγηθείτε στον κατάλογο **`lab8`**. Θα πρέπει να βρίσκονται μέσα τα αρχεία **`lab8a.c`** και **`lab8b.c`**
- Κατασκευάστε να κατάλογο με όνομα lab8_επώνυμο1_ΑΕΜ1_επώνυμο2_ΑΕΜ2 (ΜΕ ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ) όπου επώνυμο1, ΑΕΜ1 αναφέρονται στο ένα μέλος της ομάδας και επώνυμο2, ΑΕΜ2 στο δεύτερο μέλος. Για παράδειγμα, αν τα δύο μέλη της ομάδας ήταν οι Ντουφεξή με ΑΕΜ 01234 και Θάνος με ΑΕΜ 05678, θα γράφατε:

  `mkdir lab8_Doufexi_01234_Thanos_05678`
- Αντιγράψτε τα αρχεία C μέσα στον κατάλογο lab8_επώνυμο1_ΑΕΜ1_επώνυμο2_ΑΕΜ2
- Πακετάρετε τον κατάλογο (compress here as tar.gz).
- Το ένα μέλος της ομάδας:
  - μπαίνει στο λογαριασμό του στο eclass,
  - πηγαίνει στην ενότητα Εργασίες,
  - κάνει click στην εργασία lab8 για το συγκεκριμένο τμήμα
  - ΠΡΟΣΟΧΗ!!!! Στο πεδίο Ομάδα Χρηστών επιλέγει τη σωστή ομάδα (με βάση τα ονόματα των μελών)
  - Ανεβάζει το .tar.gz αρχείο μέσω του Browse και πατά Αποθήκευση.
  - ΠΡΟΣΟΧΗ!!!! Επιβεβαιώνει ότι ανέβασε το σωστό αρχείο.