

Εργαστήριο #4
3/11/2015

I. ΖΕΣΤΑΜΑ (I)

Να γραφτεί ένα πρόγραμμα που ζητά από τον χρήστη το μέγεθος ενός πίνακα, για αποθήκευση ακεραίων. Στη συνέχεια, δεσμεύει δυναμικά χώρο για τον πίνακα αυτό, τον γεμίζει με τυχαίους ακεραίους, και υπολογίζει τον μέσο όρο τους. Στην τελική εκτύπωση, να τυπώνετε τα στοιχεία του πίνακα και τον μέσο όρο τους.

Για να χρησιμοποιήσετε (ψευδο)τυχαίους αριθμούς, πρέπει να κάνετε:

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

Επιπλέον, κάπου στην αρχή της main() θα πρέπει να έχετε την εντολή:

```
srand(time(NULL));
```

Από κει και στο εξής, για να παίρνετε τυχαίους αριθμούς από το 0 μέχρι το MAXNUM-1, αρκεί να κάνετε:

```
num = rand() % MAXNUM;
```

II. ΖΕΣΤΑΜΑ (II) – ΣΥΜΠΛΗΡΩΣΤΕ ΤΑ ΚΕΝΑ

Δίνεται το πρόγραμμα lab4-2-fill.c το οποίο δεν είναι ολοκληρωμένο και πρέπει να το ολοκληρώσετε στα σημεία που αναγράφεται σε σχόλια 'FILL HERE'. Σκοπός του είναι να ζητά από τον χρήστη το μέγεθος ενός πίνακα, τον οποίο δεσμεύει δυναμικά και γεμίζει με κάποιον αριθμό. Στη συνέχεια ζητά από τον χρήστη ένα νέο μέγεθος για τον πίνακα και του το αλλάζει, φροντίζοντας να μηδενίσει τα επιπλέον στοιχεία, εφόσον το νέο μέγεθος είναι μεγαλύτερο από το παλιό.

III. ΑΠΟ ΤΗΝ ΠΕΡΣΙΝΗ ΠΡΟΟΔΟ

Να γραφτεί πρόγραμμα στο οποίο η main() δέχεται ορίσματα και κάνει τα εξής:

- Μετρά και αθροίζει τα μήκη όλων των ορισμάτων της.
- Δεσμεύει χώρο μνήμης ίσο με τα παραπάνω άθροισμα (συν 1) για αποθήκευση νέας συμβολοσειράς.
- Αντιγράφει στον χώρο που δεσμεύτηκε όλα τα ορίσματά της, συνενώνοντάς τα (δηλαδή «κολλάει» το ένα μετά το άλλο).
- Τυπώνει τη νέα συμβολοσειρά.

IV. ΤΡΙΓΩΝΟ PASCAL

Το γνωστό τρίγωνο του Pascal είναι μια τριγωνική διάταξη των δυωνυμικών συντελεστών. Για παράδειγμα, για $n = 5$ έχει ως εξής:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Πρόκειται για έναν κάτω τριγωνικό πίνακα n γραμμών όπου τα στοιχεία της πρώτης στήλης (στήλη 0) και της διαγωνίου είναι ίσα με 1, ενώ οποιοδήποτε άλλο στοιχείο στη γραμμή i και στη στήλη j ισούται με το άθροισμα του ακριβώς από πάνω στοιχείου (δηλ. στη γραμμή $i - 1$, στήλη j) και του από πάνω και αριστερά (δηλ. στη γραμμή $i - 1$, στήλη $j - 1$).

Ζητείται να φτιάξετε συνάρτηση `int **pascal(int n);` η οποία θα κατασκευάζει δυναμικά το τρίγωνο του Pascal με n γραμμές και θα το επιστρέφει. Κάθε γραμμή του πίνακα θα πρέπει να έχει χώρο για ακριβώς όσα στοιχεία χρειάζονται (δηλ. η γραμμή i θα πρέπει να έχει $i + 1$ στοιχεία).

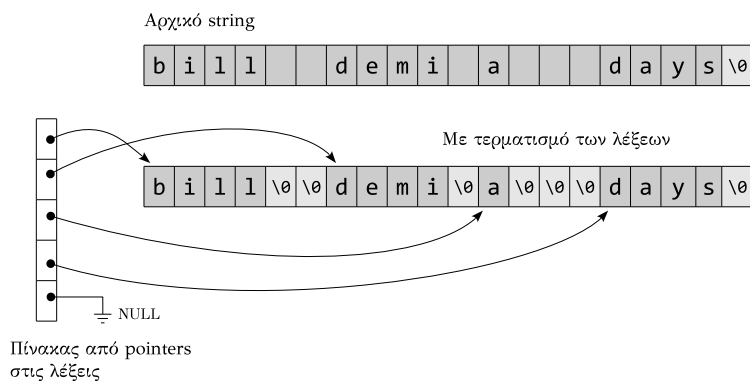
Η `main()` θα περιμένει ως μοναδικό όρισμα (`... argc, argv`) το n , θα καλεί την `pascal()` για να φτιαχτεί το τρίγωνο, θα το τυπώνει στην οθόνη, και θα απελευθερώνει την μνήμη που δεσμεύτηκε.

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: TOKENS

Δοθείσης μίας συμβολοσειράς, θα πρέπει να βρείτε τις λέξεις που περιέχει. Οι λέξεις θεωρείται ότι χωρίζονται μεταξύ τους με ένα ή περισσότερα κενά (π.χ. spaces, tabs, newlines). Πιο συγκεκριμένα,

- θα πρέπει να διατρέξετε μία φορά τη συμβολοσειρά να βρείτε πόσες λέξεις περιέχει,
- να δημιουργήσετε (με `malloc()`) έναν πίνακα από όσους pointers χρειάζονται, οι οποίοι να δείχνουν στην αρχή της κάθε λέξης,
- κάθε κενός χαρακτήρας της συμβολοσειράς θα πρέπει να αντικατασταθεί από το `\0`
- και τέλος, ο πίνακας πρέπει να έχει έναν παραπάνω δείκτη ο οποίος θα δείχνει σε NULL, ώστε να σηματοδοτείται το τέλος των λέξεων.

Ένα παράδειγμα δίνεται στο παρακάτω σχήμα.



Θα πρέπει να υλοποιήσετε μία συνάρτηση

```
char **string2words(char *str);
```

η οποία κάνει την παραπάνω διαδικασία και επιστρέφει τον πίνακα με τους δείκτες που δημιούργησε.

Τη συμβολοσειρά θα πρέπει να τη διαβάζει η `main()` από τον χρήστη με χρήση της `fgets()`. Τέλος, σε ξεχωριστή συνάρτηση να κάνετε εκτύπωση των λέξεων, μία σε κάθε γραμμή.