

### Προγραμματιστικές Ασκήσεις, Φυλλάδιο 3

Εκφώνηση: 27/4/2005

Παράδοση: 27/5/2005

#### **Άσκηση 1<sup>η</sup>**

Υλοποιήστε το πρόγραμμα της άσκησης 4, της προηγούμενης σειράς ασκήσεων, με structs. Συγκεκριμένα, γράψτε ένα πρόγραμμα που διαβάζει από ένα αρχείο στατιστικά καλαθοσφαιριστών για ένα αγώνα. Στην αρχή το πρόγραμμα ρωτά (από το πληκτρολόγιο) για πόσους αθλητές θα δώσουμε στοιχεία. Στη συνέχεια, το πρόγραμμα διαβάζει τα στοιχεία των αθλητών από ένα αρχείο ονόματι `Input.Ascii`. Για κάθε παίκτη χωριστά, καταχωρούνται τα εξής στοιχεία: (α) ο αριθμός του παίκτη (μεταξύ 0 και 99), (β) μικρό όνομα και επίθετο του παίκτη (το πολύ 20 χαρακτήρες έκαστο), (γ) οι πόντοι που έβαλε ο παίκτης, (δ) τα φάουλ που έκανε ο παίκτης και (ε) τα λεπτά συμμετοχής του παίκτη στον αγώνα. Ορίστε ένα struct για να καταχωρείται αυτή η πληροφορία στη μνήμη.

Στα ονόματα και τα επίθετα επιτρέπεται να υπάρχουν μικρά και κεφαλαία γράμματα με οποιαδήποτε σειρά. Π.χ.,

44 DraZEn PeTRovic 35 2 38

Αφού το διάβασμα από το αρχείο έχει ολοκληρωθεί, το πρόγραμμα οφείλει να έχει κρατήσει όλα τα στοιχεία που του δόθηκαν. Στη συνέχεια, το πρόγραμμα:

- ο εκτυπώνει τα στοιχεία όλων των παικτών, διαμορφωμένα ως εξής: το μικρό όνομα ξεκινά από κεφαλαίο, με τα υπόλοιπα γράμματα μικρά, το επίθετο είναι όλο με κεφαλαία γράμματα και όλα τα στοιχεία χωρίζονται με tab
- ο εκτυπώνει το μέσο όρο πόντων, φάουλ και λεπτών συμμετοχής από όλους τους αθλητές
- ο εκτυπώνει τα στοιχεία των αθλητών που το όνομα ή το επίθετό τους περιέχει τη συμβολοσειρά 'Larry', ασχέτως κεφαλαίων ή μικρών γραμμάτων
- ο εκτυπώνει τα στοιχεία των αθλητών που το επίθετό τους αρχίζει από φωνήεν
- ο μεταξύ όλων των ονομάτων και επιθέτων εκτυπώνει αυτό με το μεγαλύτερο μήκος

Υλοποιήστε μια συνάρτηση για κάθε μία από τις παραπάνω λειτουργίες. Όλες οι συναρτήσεις, εκτός από την οθόνη, τυπώνουν τα αποτελέσματα και σε ένα αρχείο αποτελεσμάτων ονόματι `Report.Ascii`.

#### **Άσκηση 2<sup>η</sup>**

Μετατρέψτε το πρόγραμμα της άσκησης 1, ώστε να μη ρωτά αρχικά πόσους παίκτες θα δώσουμε. Έτσι, αντί να αποθηκεύονται οι πληροφορίες σε πίνακα από struct, αποθηκεύονται σε μια απλή συνδεδεμένη λίστα.

#### **Άσκηση 3<sup>η</sup>**

Κατασκευάστε ένα πρόγραμμα το οποίο θα δέχεται σαν είσοδο από τη γραμμή εντολών το όνομα ενός αρχείου που θα περιέχει ένα απλό header file σε C. Το header file θα έχει την εξής γενική δομή.

1. Αρχικά θα υπάρχουν μία ή περισσότερες εντολές `include` της μορφής:  
`#include <filename.h>`
2. Οι εντολές `include` θα ακολουθούνται από το σχόλιο `/* end of include */` το οποίο ακολουθεί σε ξεχωριστή γραμμή.
3. Εν συνεχεία θα υπάρχουν μία ή περισσότερες γραμμές οι οποίες θα περιέχουν ένα ή περισσότερα πρωτότυπα συναρτήσεων η κάθε μία. Κάθε συνάρτηση μπορεί να δέχεται οσαδήποτε ορίσματα.
4. `KENES GRAMMES DEN TH YPARCHOYN METAEY TON PARAPANΩ`

Το πρόγραμμα θα διαβάζει το αρχείο με χρήση δύο συναρτήσεων οι οποίες ταυτόχρονα θα εκτελούν τις κάτωθι λειτουργίες.

1. `void find_includes(LIST *includes, FILE *fp)`: η συνάρτηση αυτή θα αποθηκεύει σε μια απλά συνδεδεμένη λίστα `includes` τα ονόματα των αρχείων που γίνονται `include` στο πρόγραμμα.

2. `void find_prototypes(LIST *prototypes, FILE *fp)`: η συνάρτηση αυτή θα διαβάζει τα πρωτότυπα των συναρτήσεων και κάθε πρωτότυπο μιας συνάρτησης θα αποθηκεύεται σε ένα κόμβο της λίστας `prototypes`.

Η λίστα σας θα έχει την εξής δομή:

```
struct list_node {
    char *program_code;
    struct list_node *next;
};
typedef struct list_node LIST_NODE;

struct list {
    LIST_NODE *top;
};
typedef struct list LIST;
```

**Υπόδειξη:** χρησιμοποιήστε τη συνάρτηση `strtok` για να βρείτε (α) τα αρχεία που γίνονται `include` και (β) τα πρωτότυπα των συναρτήσεων. Σκεφτείτε κάθε φορά, τι θα χρησιμοποιήσετε ως `tokens`. Θυμηθείτε ότι η `strtok` κατασκευάζει ένα **νέο** `string` και επιστρέφει ένα `pointer` στην αρχική του θέση, και εκμεταλλευθείτε το σε σχέση με το πεδίο `program_code`.

Να θυμάστε, όμως, ότι αυτό είναι μία από τις λίγες περιπτώσεις που δεν λαμβάνουμε χώρο εξ' αρχής για να αποθηκεύσουμε `strings`.

Παράδειγμα αρχείου εισόδου, έστω με όνομα **ask3-3.h**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
/* end of include */
void f(); int g(int a, char *X);
struct Person * q(int a);
```