

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΔΙΚΑΣΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ 2023

Τρίτη Εργαστηριακή Άσκηση – έκδοση 3.0

Βασικές έννοιες

Μπορούμε να χειριστούμε ένα αλφαριθμητικό ως πίνακα χαρακτήρων.

```
#include <stdio.h>
#include <string.h>
int main ( ) {
    char test[9 + 1];
    scanf("%s", test);
    printf("word: %s", test);
    return 0;
}
```

Τα στοιχεία του αλφαριθμητικού είναι τύπου χαρακτήρα:

```
#include <stdio.h>
#include <string.h>
int main ( ) {
    char test[9 + 1];
    int i;

    scanf("%s", test);
    for (i=0; test[i]!=0; i++) {
        printf("%c\n", test[i]);
    }
    return 0;
}
```

Για να επεξεργαστούμε αλφαριθμητικά, χρησιμοποιούμε συναρτήσεις της βασικής βιβλιοθήκης, όπως strcmp, strcpy, strlen

```
#include <stdio.h>
#include <string.h>
int main ( ) {
    char test[9 + 1];
    char dok[9 + 1];
    unsigned int len;
    strcpy(dok, "hello");
    while (scanf("%s", test), strcmp(test, dok)) {
        printf("word in: %s ", test);
        len = strlen(test);
        printf("length: %u chars\n", len);
    }
    return 0;
}
```

Για να χειριστούμε περισσότερες από μια λέξεις υπάρχουν πολλοί τρόποι. Ο απλούστερος είναι πίνακας δύο διαστάσεων

```
#include <stdio.h>
#include <string.h>
#define W 5
int main ( ) {
    char words[W][9 + 1] ;
    char test[9 + 1];
    char endword[9 +1] = "end";
    int i=0;
    while (scanf("%s", test), strcmp(test,endword)) {
        strcpy(words[i++], test);
        printf("%s\n", test);
    }
    for (i=0; i<W; i++) {
        printf("%d %s\n", i, words[i]);
    }
    return 0;
}
```

Για το δίδωρο στο εργαστήριο:

Τρέξτε όλα τα προηγούμενα προγράμματα.

Τροποποιήστε το ανωτέρω πρόγραμμα δημιουργώντας τις ακόλουθες εκδόσεις

Έκδοση 1_0: (step1_0.c) να μην είναι δυνατό το πλήθος των λέξεων να υπερβείτα όρια του πίνακα.

Έκδοση 1_1: (step1_1.c) να τυπώνονται οι λέξεις που δόθηκαν από το πληκτρολόγιο μόνο (αν είναι λιγότερες από W).

Έκδοση 1_2: (step1_2.c) να δημιουργηθεί νέος πίνακας, ίδιων διαστάσεων με τον αρχικό, στον οποίο οι λέξεις να τοποθετηθούν ανεστραμμένες αφού τους αφαιρεθούν τα φωνήεντα, a, e, i, o, u. Το 'γ' να θεωρηθεί φωνήεν μόνο αν **δεν** βρίσκεται στην αρχή μιας λέξης.

Αν ο ένας πίνακας περιέχει

Hello

Yellow

Bye

Ο δεύτερος πίνακας θα περιέχει

llH

wllY

B

Έκδοση 1_3: (step1_3.c) στον δεύτερο πίνακα, όλες οι λέξεις να γραφούν με κεφαλαία. Σκεφτείτε αν εξυπηρετεί η συνάρτηση toupper() με πρότυπο στο ctype.h

Έκδοση 1_4: (step1_4.c) να παρεμβληθεί, μετά τη δημιουργία του δεύτερου πίνακα μianέα λέξη σε αυτόν, στην πρώτη θέση μετακινώντας αντίστοιχα τις υπόλοιπες. Σε περίπτωση που ο πίνακας έχει ήδη W λέξεις, η τελευταία θα απορρίπτεται. Η νέα λέξη θα δίνεται από το πληκτρολόγιο.

Έκδοση 1_5: (step1_5.c) να παρεμβληθεί μια νέα λέξη στον δεύτερο πίνακα σε θέση που επιλέγεται κατά την ώρα της εκτέλεσης του προγράμματος. (Ζητείται είσοδος από τον χρήστη για θέση και νέα λέξη). Σε περίπτωση που ο πίνακας έχει ήδη W λέξεις, η τελευταία λέξη θα απορρίπτεται.

Το αντικείμενο της άσκησης είναι η ανάλυση κειμένου. Ζητείται να γράψετε ένα πρόγραμμα στο οποίο θα προσφέρει στον χρήστη τις εξής **βασικές επιλογές** (τουλάχιστον):

1. Εισαγωγή κειμένου (αρχικά από το πληκτρολόγιο σε μεταγενέστερες εκδόσεις από αρχείο χαρακτήρων ASCII),
2. Εισαγωγή λεξιλογίου,
3. Λειτουργία διόρθωσης κειμένου,
4. Αποθήκευση κειμένου,
5. Υπολογισμός στατιστικών του κειμένου,
6. Έξοδος από το πρόγραμμα.

Για τη λειτουργία διόρθωσης: Το πρόγραμμα ελέγχει αν κάθε λέξη του κειμένου βρίσκεται στο λεξιλόγιο. Σε περίπτωση που μία λέξη του κειμένου δεν βρίσκεται στο λεξιλόγιο, το πρόγραμμα θα εμφανίζει τη λέξη αυτή, και θα προσφέρει στον χρήστη τις ακόλουθες επιλογές:

- α) αντικατάσταση της λέξης αυτής στο κείμενο, με άλλη, νέα, λέξη την οποία θα ζητά από τον χρήστη,
- β) προσθήκη της λέξης στο λεξιλόγιο,
- γ) καμία ενέργεια σχετική με τη συγκεκριμένη λέξη και συνέχιση της διαδικασίας ελέγχου για το υπόλοιπο κείμενο, και
- δ) εγκατάλειψη της λειτουργίας διόρθωσης και επιστροφή στις βασικές επιλογές του προγράμματος.

Διευκρινίζεται ότι στην ανωτέρω περίπτωση (α) θα πρέπει να ελέγχεται εάν η νέα λέξη περιλαμβάνεται στο λεξιλόγιο.

Για τη λειτουργία υπολογισμού στατιστικών του κειμένου: Το πρόγραμμα θα εκτελεί τα εξής:

- α) θα υπολογίζει από πόσες λέξεις και από πόσους χαρακτήρες αποτελείται το κείμενο, (δύο περιπτώσεις για τους χαρακτήρες, με κενά και χωρίς),
- β) θα υπολογίζει από πόσες **διαφορετικές** λέξεις αποτελείται το κείμενο,
- γ) θα απεικονίζει με τρόπο της επιλογής σας, το ιστόγραμμα αριθμού λέξεων συναρτήσει του πλήθους των γραμμάτων τους, δηλ. πόσες λέξεις του κειμένου έχουν μήκος ένα χαρακτήρα, πόσες δύο χαρακτήρες, κτλ.
- δ) τα αποτελέσματα της ανάλυσης (διαδικασίες α έως και γ) θα σώζονται σε κατάλληλο αρχείο ASCII, με όνομα επιλογής του χρήστη.

Εκτός των λειτουργιών αυτών το πρόγραμμα θα πρέπει να παρέχει στον χρήστη βασικές επιλογές όπως, ενδεικτικά, η εισαγωγή των ονομάτων των αρχείων που διαχειρίζεται το πρόγραμμα και η εμφάνιση στην οθόνη των αποτελεσμάτων του υπολογισμού στατιστικών, **εφόσον** έχουν υπολογιστεί.

ΟΔΗΓΙΕΣ ΚΑΙ ΕΡΩΤΗΜΑΤΑ

1. Ποιες βασικές διεργασίες θα πρέπει να υλοποιεί το πρόγραμμά σας;
2. Να γράψετε (σύντομη) λεκτική περιγραφή της λειτουργίας του προγράμματος, σε αρχείο pdf. Αν θέλετε, να χρησιμοποιήσετε το <https://www.sourcetrail.com/> ώστε να δημιουργήσετε με αυτό μια γραφική αποτύπωση της τελικής έκδοσης του προγράμματός σας και να την περιλάβετε στο pdf. (Ένα σχήμα, αφού ολοκληρώσετε τα επόμενα βήματα).
3. Στη συνέχεια να γράψετε το πρόγραμμα. Το πρόγραμμα θα πρέπει να είναι οργανωμένο ως εξής: Να έχει συνάρτηση main με βρόχο επανάληψης κατά τον οποίο θα εισάγεται η επιλογή του χρήστη (η οποία θα δίνεται σε ξεχωριστή συνάρτηση) και ανάλογα με την επιλογή, θα τερματίζεται ο βρόχος ή θα καλείται αντίστοιχη συνάρτηση λειτουργίας.
4. Η ανάπτυξη του προγράμματος θα πρέπει να γίνει με τις εξής φάσεις:

Έκδοση 2_0: (step2_0.c) Υλοποιείται η οργάνωση του προγράμματος ως κλήση προσωρινών συναρτήσεων. (Ενδεικτικά, οι συναρτήσεις απλώς τυπώνουν το όνομά τους, χωρίς να υλοποιούν την τελική λειτουργία τους.)

Έκδοση 2_1: (step2_1.c) Το πρόγραμμα δεν χειρίζεται αρχεία. Το κείμενο εισάγεται ως ξεχωριστές λέξεις από το πληκτρολόγιο, μέχρις ότου να διαβαστεί η λέξη `"*T*E*L*O*S*"`. Οι τύποι δεδομένων που χρησιμοποιεί το πρόγραμμά μας είναι πίνακες χαρακτήρων σταθερού μεγέθους. Δηλαδή θέτουμε προκαθορισμένο όριο μεγέθους στους πίνακες.

Έκδοση 2_2: (step2_2.c) Το κείμενο εισάγεται από αρχείο κειμένου. Αντικαθιστούμε τις σχετικές συναρτήσεις.

Έκδοση 2_3: (step2_3.c) Τα στατιστικά αποθηκεύονται σε αρχείο.

Έκδοση 2_4: (step2_4.c) Υλοποιείται η λειτουργία διόρθωσης λέξης, όπως παρουσιάστηκε νωρίτερα.

Ζητείται να αποφύγετε τη χρήση καθολικών (global) μεταβλητών και να προσπαθήσετε να χρησιμοποιήσετε τον μεγαλύτερο αριθμό συναρτήσεων.

ΠΑΡΑΔΟΤΕΑ

Το σύνολο των παραδοτέων του πρώτου βήματος που θα ολοκληρώσετε στο εργαστήριο τοποθετείται σε ένα συμπιεσμένο αρχείο το οποίο και ανεβάζετε στο eclass με το τέλος του εργαστηρίου.

Τα παραδοτέα του δεύτερου βήματος θα παραδοθούν μέσω του eclass πριν τη σχετική ημερομηνία.