



3^η Εργασία

Διαδικαστικά

Η εργασία είναι **αυστηρά ατομική** και αποτελεί την 3^η από τις 5 εργασίες του μαθήματος. Ως 5^η εργασία θα υπολογιστεί η συμμετοχή στη διόρθωση μιας εργασίας. Τα διαδικαστικά που αφορούν τις εργασίες αναφέρονται αναλυτικά στις πληροφορίες του μαθήματος στο eClass. **Αντιγραφή σε κάποια εργασία συνεπάγεται μηδενισμό σε όλες τις εργασίες αυτού του έτους.**

Όλες οι εργασίες θα παραδοθούν αυστηρά μέσω eClass.

Η 3^η εργασία έχει καταληκτική ημερομηνία και ώρα παράδοσης **Πέμπτη 13/12/2018** και ώρα **23:30** (πείτε στον εαυτό σας ότι το σύστημα κλείνει 11 το βράδυ και ότι η μισή ώρα είναι για να μην τύχει κάτι). **Καμία εργασία δεν θα γίνει δεκτή μετά τη λήξη της προθεσμίας¹.**

Ζητούμενο

Οι δύο υλοποιήσεις σε ANSI C, που ακολουθούν εκτελούν την ίδια λειτουργία: παίρνουν σαν είσοδο δύο λέξεις και ελέγχουν εάν η μία είναι αναγραμματισμός της άλλης (μια λέξη θεωρείται αναγραμματισμός μιας άλλης αν περιέχει ακριβώς τα ίδια γράμματα και ίδιο πλήθος εμφανίσεων για κάθε γράμμα). Θεωρούμε ότι οι λέξεις περιέχουν πεζά γράμματα του αγγλικού αλφαβήτου και το μέγιστο μήκος τους είναι 30 χαρακτήρες.

1. Κατασκευάστε **για κάθε μία από τις παραπάνω τρεις ρουτίνες** ένα αναλυτικό πίνακα όπου θα παρουσιάζονται οι διάφοροι τελεστές και τα έντελα, καθώς και το πλήθος των εμφανίσεών τους. Ο πίνακας αυτός θα πρέπει να έχει την παρακάτω μορφή.

Τελεστές	Αριθμός εμφανίσεων	Έντελα	Αριθμός εμφανίσεων
$n_1=$	$N_1=$	$n_2=$	$N_2=$

2. Υπολογίστε για κάθε μία από τις τρεις ρουτίνες:

- το λόγο του εκτιμητή μήκους προς το μήκος προγράμματος του Halstead (N_{est}/N),
- το επίπεδο προγράμματος του Halstead (L),
- το επίπεδο γλώσσας του Halstead (λ),
- το λόγο αριθμού γραμμών σχολίων προς τον αριθμό φυσικών γραμμών κώδικα (Lines of Comments / Physical Lines of Code).

¹ Αυτό είναι κάτι που το τηρώ αυστηρά και δεν θα παρεκκλίνω ποτέ, άρα μην στείλετε εργασία έστω και 1 λεπτό μετά τη λήξη της προθεσμίας με e-mail.



3. Για τον κώδικα της Β' υλοποίησης υπολογίστε μια συνολική τιμή για κάθε μία από τις παραπάνω μετρικές εξετάζοντας τα εξής σενάρια:

- Σ1. οι συνολικές τιμές στις μετρικές υπολογίζονται από το μέσο όρο των τιμών τους σε κάθε ρουτίνα
- Σ2. οι συνολικές τιμές στις μετρικές υπολογίζονται από το σταθμισμένο μέσο όρο των τιμών τους σε κάθε ρουτίνα, με βάση το N

Ποιο σενάριο θεωρείτε καταλληλότερο; Τεκμηριώστε την απάντησή σας, λαμβάνοντας υπόψη και τη γενικότερη περίπτωση εφαρμογής των μετρικών αυτών σε κώδικα που περιλαμβάνει πολλές ρουτίνες.

4. Με βάση το σενάριο που επιλέξατε να συγκρίνετε τις δύο υλοποιήσεις Α και Β και να σχολιάσετε το αποτέλεσμα ερμηνεύοντας τις τιμές των μετρικών.



Α' υλοποίηση

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char a[30], b[30];
    int first[26] = {0}, second[26] = {0}, c = 0;
    int flag;

    printf("Give the first word \n");
    gets(a);    // είσοδος πρώτης λέξης από το χρήστη
    printf("Give the second word \n");
    gets(b);    // είσοδος δεύτερης λέξης από το χρήστη
    // ορίζω δυο πίνακες για τα γράμματα κάθε λέξης, π.χ. στο
    // first[0] θα αποθηκεύω το πλήθος εμφανίσεων του γράμματος 'a'
    // στην πρώτη λέξη και στο second[1] το πλήθος εμφανίσεων του
    // γράμματος 'b' στη δεύτερη λέξη
    while (a[c] != '\0')
    {
        // αυξάνω κατά ένα την αντίστοιχη θέση του πίνακα
        // για την πρώτη λέξη
        first[a[c]-'a']++;
        c++;
    }
    c = 0;
    while (b[c] != '\0')
    {
        // αυξάνω κατά ένα την αντίστοιχη θέση του πίνακα για την 2η λέξη
        second[b[c]-'a']++;
        c++;
    }
    for (c = 0; c < 26; c++)
    {
        // αν οι δυο πίνακες που κρατάνε το πλήθος των εμφανίσεων
        // είναι διαφορετικοί τότε δεν είναι αναγραμματισμός
        if (first[c] != second[c])
        {
            flag=0;
            break;
        }
        else
            flag=1;
    }
    if (flag == 1)
        printf("The words ARE anagrams of one another");
    else
        printf("The words ARE NOT anagrams of one another");
    return 0;
}
```



Β' υλοποίηση

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int comp(const void *a, const void *b)
{
    /* const char *pa = a;
    const char *pb = b;
    return
        (*pa > *pb) ? 1 :
        (*pa < *pb) ? -1 :
        0; */
    return (*(char*)a - *(char*)b);
}

int main()
{
    char s1[30], s2[30];
    printf("Give the first word \n");
    gets(s1);
    printf("Give the second word \n");
    gets(s2);
    //ταξινομούνται αλφαβητικά οι χαρακτήρες των λέξεων
    qsort(s1, strlen(s1), 1, comp);
    qsort(s2, strlen(s2), 1, comp);
    printf(strcmp(s1, s2) ? "The words ARE NOT anagrams of one another" :
    "The words ARE anagrams of one another");
    return 0;
}
```

Η τελευταία εντολή printf στην Β' υλοποίηση καταλαμβάνει μία γραμμή. Τις τρεις γραμμές με τις εντολές #include τις προσμετράμε στην main, σε όποιες μετρικές χρειάζεται.

Σημείωση: Στους υπολογισμούς σας για τις μετρικές του Halstead να μην προσμετρήσετε την επικεφαλίδα κάθε ρουτίνας και τα εξωτερικά { } της ρουτίνας, αλλά να προσμετρήσετε τις δηλώσεις μεταβλητών.

ΣΗΜΑΝΤΙΚΟ:

Διαβάστε αυτές τις λίγες γραμμές για να μην χάνετε εσείς μονάδες και εγώ την ψυχραιμία μου! Για την εργασία παραδώστε **ΜΟΝΟ ένα αρχείο pdf με τη λύση σας και τίποτε άλλο!** Το αρχείο αυτό ονομάστε το με το όνομά σας (π.χ. Xenos_m.pdf), αλλά μέσα στο αρχείο δεν θα πρέπει να υπάρχει τίποτε απολύτως που να αναφέρει το όνομά σας. Αυτό περιλαμβάνει και τα metadata του αρχείου, δηλαδή να σβήσετε κάθε προσωπική πληροφορία και από τα properties του αρχείου (π.χ. όνομα). Σε περίπτωση που το αρχείο σας έχει οποιαδήποτε πληροφορία για εσάς, είτε στο κείμενο είτε στα metadata θα βαθμολογηθεί κανονικά, αλλά με -30% του βαθμού ως ποινή.

Μην αφήνετε την εργασία για τελευταία στιγμή και **ΜΗΝ εμπλακείτε σε διαδικασίες που μπορεί να σας φέρουν σε δύσκολη θέση.**