

HY-100: Εισαγωγή στην Επιστήμη Υπολογιστών

4η σειρά ασκήσεων

Οδηγίες

Για τη μεταγλώττιση των προγραμμάτων που ζητούνται θα πρέπει να χρησιμοποιήσετε το gcc με τις παρακάτω παραμέτρους

```
gcc -Wall
```

και το πρόγραμμα σας να κάνει compile χωρίς warnings.

ΠΡΟΣΟΧΗ: Οι ασκήσεις θα βαθμολογηθούν με αυτόματο τρόπο, οπότε θα πρέπει να υπακούν πιστά την εκφώνηση (ακόμη και τα ονόματα των αρχείων).

Για να ελέγξετε την άσκηση σας, εκτελέστε τις εντολές

```
cd hw4/  
sh test.sh hw4 tests
```

όπου hw4 είναι το όνομα του εκτελέσιμου προγράμματος σας και tests ο φάκελος με τα test. Κάθε test αποτελείται από ένα αρχείο με κατάληξη .in, που περιέχει την είσοδο για το πρόγραμμα σας, και το αντίστοιχο .out αρχείο που περιέχει την αναμενόμενη έξοδο του προγράμματος.

Οι ασκήσεις θα πρέπει να τρέχουν και να περνούν τα tests στα μηχανήματα της σχολής, όπου και θα βαθμολογηθούν.

Για κάθε test που περνάει το πρόγραμμα σας θα τυπώνεται **PASS**, αλλιώς **FAIL**.

1 Μέτρηση αριθμού διαφορετικών λέξεων σε τυχαίο κείμενο

Γράψτε ένα πρόγραμμα (hw4a.c) που θα διαβάζει τυχαίο κείμενο από ένα αρχείο, του οποίου το path θα περνάτε ως command line argument στην main, και για κάθε διαφορετική λέξη στο κείμενο θα τυπώνει πόσες φορές υπάρχει μέσα σε αυτό.

Η άσκηση σας θα πρέπει:

- να ελευθερώνει την μνήμη που δέσμευσε δυναμικά στο τέλος της εκτέλεσης με τη συνάρτηση free όπως και να κλείνει το αρχείο με την fclose

Συνίσταται:

- η χρήση της calloc αντί της malloc η οποία αρχικοποιεί τη μνήμη που δεσμεύει
- η χρήση realloc για να μεγαλώνετε τη μνήμη που έχετε δεσμεύσει

Για χάρην ευκολίας θεωρήστε ότι:

- όλες οι λέξεις χωρίζονται μεταξύ τους είτε με κενό είτε με newline
- οι ειδικοί χαρακτήρες πέραν κενού και newline θεωρούνται κομμάτι μίας λέξης

- ίδιες λέξεις που ξεκινούν η μία με μικρό και η άλλη με κεφαλαίο θεωρούνται διαφορετικές λέξεις

Για παράδειγμα, για την ακόλουθη είσοδο

```
Computer science is the study of computation, automation, and information.[1]
Computer science spans theoretical disciplines (such as algorithms, theory
of computation, information theory, and automation) to practical disciplines
(including the design and implementation of hardware and software).[2][3][4]
Computer science is generally considered an area of academic research and
distinct from computer programming.[5]
```

το πρόγραμμά σας θα πρέπει να τυπώνει

```
Computer 3
science 3
is 2
the 2
study 1
of 4
computation, 2
automation, 1
and 5
information.[1] 1
spans 1
theoretical 1
disciplines 2
(such 1
as 1
algorithms, 1
theory 1
information 1
theory, 1
automation) 1
to 1
practical 1
(including 1
design 1
implementation 1
hardware 1
software).[2][3][4] 1
generally 1
considered 1
an 1
area 1
academic 1
research 1
distinct 1
from 1
computer 1
programming.[5] 1
```

Για να ελέγξετε την άσκησή σας χρησιμοποιήστε τον κατάλογο hw4/hw4a-tests που δίνεται:

```
sh test.sh hw4a hw4a-tests
```

2 Βαθμολογίες Ονομάτων

Στη παρούσα άσκηση σας ζητείται να γράψετε ένα πρόγραμμα (hw4b.c) το οποίο θα διαβάζει ένα πλήθος από ονόματα, αποθηκευμένα σε ένα αρχείο κειμένου και στην συνέχεια θα επιστρέφει το άθροισμα των βαθμολογιών (scores) όλων των ονομάτων με τον εξής τρόπο:

- Αρχικά διαβάζουμε το αρχείο που περιέχει τα ονόματα και τα ταξινομούμε με (αύξουσα) αλφαβητική σειρά (είναι καλή πρακτική να υλοποιήσετε κάποιον αλγόριθμο που έχετε διδαχθεί στο μάθημα αντί κάποιας υπάρχουσας συνάρτησης).
- Στη συνέχεια, αθροίζουμε τις αλφαβητικές τιμές για κάθε όνομα, τις πολλαπλασιάζουμε με την αριθμητική θέση του ονόματος στον ταξινομημένο πλέον πίνακα αλφαριθμητικών και λαμβάνουμε μια βαθμολογία ονόματος (number score).
- Ζητούμε το σύνολο (άθροισμα) των βαθμολογιών όλων των ονομάτων στο δοθέν αρχείο.

Για παράδειγμα, εστω ότι μια λίστα ονομάτων ταξινομείται με αλφαβητική σειρά. Το όνομα ANNA, βαθμολογείται ως $1 + 14 + 14 + 1 = 28$ (διότι το A είναι το πρώτο γράμμα της αλφαβήτου, το N το 14ο κτλ). Εάν στην ταξινομημένη λίστα το όνομα ANNA είναι πέμπτο, τότε θα λάβει βαθμολογία (score) $28 \times 5 = 140$. Το πρόγραμμά μας θα επιστρέψει το σύνολο (άθροισμα) των βαθμολογιών όλων των ονομάτων στο δοθέν αρχείο.

Το πρόγραμμά σας θα λαμβάνει ως όρισμα το όνομα του αρχείου, κάνοντας κατάλληλη χρήση των ορισμάτων argc και argv της main συνάρτησης. Στο πέρας της εκτέλεσής το πρόγραμμά σας θα εκτυπώνει τη συνολική βαθμολογία (total score) του αρχείου, με KENO διάστημα και εντολή newline για την επόμενη εκτέλεση (προσοχή σε αυτό το βήμα ώστε να μην έχετε προβλήματα με τον autograder). Να λάβετε υπόψη την περίπτωση που το δοθέν αρχείο δεν υπάρχει. Τα αρχεία που παρέχονται στα test files είναι μορφής απλού κειμένου, κατάληξης .in (ως test input), με όλα τα αλφαριθμητικά των ονομάτων στην πρώτη γραμμή, σε κεφαλαία γράμματα.

Για παράδειγμα, έστω ένα αρχείο namesfile.txt που βρίσκεται στο ίδιο directory με το εκτελέσιμο a.out, το οποίο περιέχει μόνο 2 ονόματα

```
"RICHARD", "JENNIFER"
```

Το πρόγραμμα, αφού εκτελεστεί ως

```
./a.out namesfile.txt
```

ταξινομεί πρώτα το όνομα JENNIFER και μετά το RICHARD, και εκτυπώνει συνολικό score 165, που είναι το άθροισμα των score των δύο ονομάτων, δηλαδή στην οθόνη εμφανίζεται

```
./a.out namesfile.txt  
165
```

Φροντίστε να μην παραλείψετε την εντολή `fclose` στο τέλος του προγράμματός σας. Αναφορικά με την διαχείριση μνήμης, για διευκόλυνσή σας, ώστε να αποφύγετε `overflows` και `segmentation faults`, μπορείτε μέσω κάποιου `macro` να θεωρείσετε μέγιστο μέγεθος αρχείου 200 bytes και ότι το πλήθος των ονομάτων σε όλα τα `test files` είναι 15. Η άσκηση συνδυάζει αρκετά πράγματα που έχουν διδαχθεί όπως δείκτες, διαχείριση μνήμης, χειρισμό αρχείων και αλφαριθμητικών και ταξινόμηση, οπότε υλικό που έχετε χρησιμοποιήσει στις προηγούμενες ασκήσεις θα σας φανεί χρήσιμο.

Για να ελέγξετε την άσκησή σας χρησιμοποιήστε τον κατάλογο `hw4/hw4b-tests` που δίνεται:

```
sh test.sh hw4b hw4b-tests
```

Οδηγίες παράδοσης σειράς ασκήσεων

Παραδώστε τα αρχεία `hw4a.c` και `hw4b.c`, ακολουθώντας τις οδηγίες που βρίσκονται στη σελίδα <https://www.csd.uoc.gr/~hy100/turnin-howto-gr.html>.