

ΣΕΤ ΑΣΚΗΣΕΩΝ 3

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2016-2017

Προθεσμία: Τετάρτη 7/12/2016, 21:00

Διαβάστε πριν ξεκινήσετε

Διαβάστε όλη την εκφώνηση προσεκτικά πριν ξεκινήσετε να γράφετε κώδικα. Ολοκληρώστε κάθε ένα στάδιο πριν προχωρήσετε στο επόμενο, κι αποθηκεύστε σε ξεχωριστά αρχεία ενδιάμεσες σωστές μορφές του προγράμματός σας ώστε να μπορείτε να επανέλθετε σε αυτές αν κάνετε κάποιο λάθος.

Η εργασία αυτή μπορεί να γίνει σε ομάδες **μέχρι 2 ατόμων**. Δε χρειάζεται να είστε ομάδα με το ίδιο άτομο που είστε στο εργαστήριο. Μπορείτε να συζητάτε τις ασκήσεις με συμφοιτητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιονδήποτε τρόπο.

Ξεκινήστε νωρίς! Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ότι περιμένετε.

Εκπρόθεσμες ασκήσεις δε γίνονται δεκτές.

Οι ασκήσεις σας θα βαθμολογηθούν στα παρακάτω (χωρίς ιδιαίτερη σειρά):

- Ορθότητα
- Αναγνωσιμότητα (στοίχιση, ονόματα μεταβλητών, κτλ.)
- Σχεδιασμός προγράμματος και αποτελεσματική χρήση κατάλληλων δομών, μεταβλητών, σταθερών, συναρτήσεων κτλ.
 - Ειδικά για τις συναρτήσεις, θα πρέπει να αποφασίσετε εσείς πώς να οργανώσετε τον κώδικά σας και τι θα ήταν καλό να γίνει σε συνάρτηση. Δε θα δώσουμε ιδιαίτερες οδηγίες για κάτι τέτοιο.
- Συμμόρφωση με τις προδιαγραφές
- Αποτελεσματικά σχόλια, σύμφωνα με τους κανόνες σχολιασμού του σχετικού φυλλαδίου.

Προσέξτε ιδιαίτερα:

- Τα όρια των loops όταν κάνετε προσπέλαση πινάκων.
- Μην επαναλαμβάνετε μεγάλα κομμάτια κώδικα (π.χ. αν έχετε ίδιες εντολές και στα δύο σκέλη μιας if/else, βγάλτε τις απέξω), ή χρησιμοποιήστε συνάρτηση.
- Μη χρησιμοποιήσετε καθολικές/static μεταβλητές, goto, gets και ότι άλλο έχουμε "απαγορεύσει" στις εργασίες.
- Τις πληροφορίες [στο τέλος της εκφώνησης](#) για τη χρήση της συνάρτησης fgets.

Άσκηση 1: Διαχείριση συμβολοσειρών

Θα γράψετε ένα πρόγραμμα το οποίο αποθηκεύει τα στοιχεία και τους βαθμούς ενός αριθμού φοιτητών σε πίνακα και υπολογίζει κάποια στατιστικά. Κύριοι στόχοι της άσκησης είναι να γίνει αποτελεσματική χρήση των συναρτήσεων που σας παρέχει η γλώσσα, καθώς και να δουλέψετε με δείκτες.

Αρχείο δεδομένων

Τα στοιχεία των φοιτητών θα διαβαστούν με ανακατεύθυνση από ένα αρχείο, στο οποίο είναι γραμμένα ως εξής:

- Σε κάθε γραμμή βρίσκονται τα στοιχεία ακριβώς ενός φοιτητή.
- Μια γραμμή περιέχει τα παρακάτω:
 - Αριθμός μητρώου στη μορφή `CCCC`. Μέγιστο πλάτος 5 χαρακτήρες. Ο αριθμός δεν είναι στοιχισμένος με κάποιο τρόπο. Μπορεί κάποιοι χαρακτήρες στην αρχή ή το τέλος να είναι κενά και όχι ψηφία, αν ο αριθμός είναι μικρότερος από πενταψήφιο.
 - Ένα κενό (`' '`) που διαχωρίζει αυτή την πληροφορία από την επόμενη.
 - Ονοματεπώνυμο, στη μορφή `Επώνυμο,Μικρό`. Μέγιστο πλάτος 25 χαρακτήρες. Αν το ονοματεπώνυμο είναι μικρότερο από 25 χαρακτήρες, οι υπόλοιποι είναι κενά. Το ονοματεπώνυμο είναι πάντα στοιχισμένο αριστερά. Παρατηρήστε πως δεν υπάρχουν κενά γύρω από το κόμμα.
 - Ένα κενό (`' '`) που διαχωρίζει αυτή την πληροφορία από την επόμενη.
 - Βαθμός, στη μορφή `CC.C`. Μέγιστο πλάτος 4 χαρακτήρες. Οι βαθμοί είναι από 0 μέχρι και 10.
 - Χαρακτήρας αλλαγής γραμμής (`'\n'`).
- Η τελευταία γραμμή περιέχει έναν αρνητικό αριθμό μητρώου, ο οποίος σηματοδοτεί τέλος των δεδομένων.

Μελετήστε τα ενδεικτικά αρχεία εισόδου που θα σας δώσουμε.

Μπορείτε να υποθέσετε ότι ο μέγιστος αριθμός φοιτητών τα στοιχεία των οποίων είναι στο αρχείο είναι 100, αλλά το πρόγραμμά σας πρέπει να είναι γραμμένο έτσι ώστε να μπορεί να αλλαχθεί εύκολα αυτή η ποσότητα.

Στάδιο 1: Είσοδος δεδομένων

Δημιουργήστε ένα πίνακα δύο διαστάσεων. Το πλήθος γραμμών είναι ο μέγιστος αριθμός φοιτητών ενώ το πλήθος στηλών είναι ακριβώς όσο χρειάζεται για να χωρέσουν τα δεδομένα που περιγράφονται πιο πάνω.

Κάθε γραμμή του πίνακα πρέπει να περιέχει μια γραμμή του αρχείου, με τα επιπλέον χαρακτηριστικά:

- Δεν υπάρχει `'\n'` στο τέλος της γραμμής, αλλά μόνο το `'\0'` που θα βάλει αυτόματα η `fgets`.
- Το κενό που διαχωρίζει το ονοματεπώνυμο από το βαθμό αντικαθίσταται από `'\0'`. Αυτό το κάνουμε για να διευκολυνθούμε στη χρήση συναρτήσεων για strings αναφορικά με το ονοματεπώνυμο.

Γράψτε κώδικα που διαβάζει τα δεδομένα μια γραμμή τη φορά (με χρήση `fgets`) και τα αποθηκεύει στον πίνακα κάνοντας τις παραπάνω αλλαγές. Λάβετε υπόψη ότι πρέπει να σταματήσετε να διαβάζετε δεδομένα αν γεμίσει ο πίνακας ή αν διαβαστεί αρνητικός αριθμός μητρώου.

Αφού αποθηκεύσετε τα περιεχόμενα στον πίνακα, **γράψτε κώδικα** που για κάθε φοιτητή εκτυπώνει:

- Το πρώτο γράμμα του μικρού του ονόματος
- Μια τελεία κι ένα κενό.
- Το επώνυμό του κι ένα κενό
- Τον αριθμό μητρώου του κι ένα κενό
- Το βαθμό του με ένα δεκαδικό ψηφίο και χαρακτήρα αλλαγής γραμμής

Για να αποσπάσετε και να εκτυπώσετε αυτές τις τιμές κάντε έξυπνη χρήση δεικτών και συναρτήσεων σχετικών με strings. Για παράδειγμα, μπορείτε να βρείτε πού ξεκινά το μικρό όνομα αν χρησιμοποιήσετε τη συνάρτηση `strchr` για να εντοπίσετε το κόμμα.

Επίσης, για να μπορεί να αλλαχθεί εύκολα το πρόγραμμά σας αν αλλάξει το μέγεθος των δεδομένων, ορίστε κατάλληλες σταθερές που σας λένε σε ποια θέση του πίνακα ξεκινά κάθε "πεδίο" (αρ. μητρώου, ονοματεπώνυμο, βαθμός) και τι μέγιστο μέγεθος έχει.

Στάδιο 2: Δημιουργία ιστογράμματος και υπολογισμός μέσου όρου

Δημιουργήστε ένα πίνακα μεγέθους 11. Σε κάθε κελί αυτού του πίνακα θα αποθηκευτεί πλήθος φοιτητών που πήραν συγκεκριμένους βαθμούς. Για την ακρίβεια, στο κελί 0 θα αποθηκευτεί το πλήθος των φοιτητών που πήραν 0. Στο κελί 1 θα αποθηκευτεί το πλήθος των φοιτητών που πήραν βαθμούς μεγαλύτερους από 0 και το πολύ ίσους με 1. Στο κελί 2 θα αποθηκευτεί το πλήθος των φοιτητών που πήραν βαθμούς μεγαλύτερους από 1 και το πολύ ίσους με 2. Ομοίως για τα υπόλοιπα κελιά του πίνακα.

Γράψτε κώδικα που διατρέχει τον διδιάστατο πίνακα δεδομένων, εξετάζει τους βαθμούς και γεμίζει κατάλληλα τον πίνακα με τα πλήθη βαθμών.

Γράψτε κώδικα που να υπολογίζει το μέσο όρο όλων των βαθμών.

Γράψτε κώδικα που εκτυπώνει ένα ιστόγραμμα στην οθόνη, το οποίο δείχνει με "γραφικό" τρόπο πόσοι φοιτητές πήραν βαθμούς σε κάθε εύρος. Για την ακρίβεια, για κάθε κελί του πίνακα το πρόγραμμα εκτυπώνει

- αριστερή παρένθεση, εκτός από την περίπτωση του 0, που εκτυπώνεται αριστερή αγκύλη [
- το κάτω όριο του εύρους (μη-συμπεριλαμβανομένου) που αντιστοιχεί σε αυτό το κελί, με πλάτος 2
- κόμμα
- ένα κενό
- το πάνω όριο του εύρους (συμπεριλαμβανομένου) που αντιστοιχεί σε αυτό το κελί (πλάτος 2)
- δεξιά αγκύλη]
- άνω-κάτω τελεία
- ένα κενό
- τόσα αστεράκια (*) όσες οι δεκάδες των φοιτητών που πήραν βαθμούς σε αυτό το εύρος
- τόσες παύλες (-) όσες οι μονάδες των φοιτητών που πήραν βαθμούς σε αυτό το εύρος.
- χαρακτήρα αλλαγής γραμμής.

Για παράδειγμα, αν υπήρχαν 23 φοιτητές που πήραν βαθμούς μεταξύ 4 και 5, τότε για αυτό το εύρος το πρόγραμμα θα εκτυπώσει:

```
( 4 , 5 ] : **---
```

Ειδικά για το πρώτο κελί του πίνακα, το κάτω και το πάνω όριο είναι μηδέν, εφόσον περιέχει το πλήθος των βαθμών που είναι μηδέν.

Αν δε βαθμολογήθηκε κανείς σε κάποιο εύρος, απλά δεν εμφανίζονται αστεράκια ή παύλες μετά την άνω-κάτω τελεία.

Χρήσιμη συνάρτηση από τη μαθηματική βιβλιοθήκη: `ceil`

Τέλος, το πρόγραμμά σας πρέπει να εκτυπώνει χαρακτήρα αλλαγής γραμμής, το μήνυμα **AVERAGE: X**, όπου X ο μέσος όρος των βαθμών με δύο δεκαδικά ψηφία, και έναν ακόμη χαρακτήρα αλλαγής γραμμής.

ΑΠΟΘΗΚΕΥΣΤΕ ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΑΣ ΣΕ ΑΡΧΕΙΟ ΜΕ ΟΝΟΜΑ hw3a.c

Άσκηση 2: Δομές

Σε αυτή την άσκηση θα γράψετε ένα πρόγραμμα παρόμοιο με το προηγούμενο, το οποίο όμως χρησιμοποιεί structs για την αναπαράσταση των δεδομένων.

Στάδιο 1: Είσοδος δεδομένων

Κατασκευάστε ένα struct με όνομα `studentData` και τα εξής πεδία:

- Αριθμός μητρώου, ακέραιος.
- Ονοματεπώνυμο, πίνακας χαρακτήρων μεγέθους 26 (25 το όνομα συν μία το `\0`)
- Βαθμός, πραγματικός αριθμός.

Κατασκευάστε ένα πίνακα από `studentData`. Το μέγεθός του πρέπει να είναι όσο το μέγιστο δυνατό πλήθος φοιτητών.

Γράψτε μια συνάρτηση η οποία παίρνει ως παράμετρο τον πίνακα, διαβάζει τα δεδομένα όπως αυτά περιγράφονται στην άσκηση 1, τα αποθηκεύει με κατάλληλο τρόπο στον πίνακα (**προσοχή**: τα ονόματα αποθηκεύονται χωρίς κενά!) κι επιστρέφει το πλήθος των φοιτητών (το οποίο θυμίζουμε δεν είναι απαραίτητα όσο και το μέγεθος του πίνακα). Τις θέσεις του πίνακα που δεν αντιστοιχούν σε φοιτητές μπορείτε να τις αρχικοποιήσετε με κάποιο κατάλληλο τρόπο που "σημαίνει" ότι είναι άκυρες, για παράδειγμα με αρνητικούς αριθμούς και κενό ονοματεπώνυμο.

Γράψτε μια συνάρτηση η οποία παίρνει ως παράμετρο τον πίνακα και το πλήθος φοιτητών και εκτυπώνει τα περιεχόμενά του όπως και στην άσκηση 1.

Καλέστε τις συναρτήσεις σας στη `main` για να αρχικοποιήσετε και μετά να εκτυπώσετε τον πίνακα.

Στάδιο 2: Ταξινόμηση

Γράψτε κώδικα στη `main` που εκτυπώνει χαρακτήρα αλλαγής γραμμής, 20 παύλες και πάλι χαρακτήρα αλλαγής γραμμής (αυτό θα μπορούσε να γίνει και μέσω συνάρτησης).

Γράψτε μια συνάρτηση με όνομα `selection_sort` η οποία παίρνει ως παράμετρο τον πίνακα και το πλήθος φοιτητών και χρησιμοποιεί τον αλγόριθμο `selection sort` για να ταξινομήσει τον πίνακα (για την ακρίβεια το μέρος του πίνακα που περιέχει έγκυρα δεδομένα) σε αύξουσα σειρά με βάση το ονοματεπώνυμο. Σε περίπτωση συνωνυμίας, χρησιμοποιήστε ως δεύτερο κριτήριο τον αριθμό μητρώου (αύξουσα σειρά).

Γράψτε μια συνάρτηση με όνομα `insertion_sort` η οποία παίρνει ως παράμετρο τον πίνακα και το πλήθος φοιτητών και χρησιμοποιεί τον αλγόριθμο `insertion sort` για να ταξινομήσει τον πίνακα (για την ακρίβεια το μέρος του πίνακα που περιέχει έγκυρα δεδομένα) σε αύξουσα σειρά με βάση το ονοματεπώνυμο. Σε περίπτωση συνωνυμίας, χρησιμοποιήστε ως δεύτερο κριτήριο τον αριθμό μητρώου (αύξουσα σειρά).

Καλέστε στη `main` τη συνάρτηση `selection_sort` για να ταξινομήσετε τον πίνακα, και κατόπιν τη συνάρτηση που εκτυπώνει τα περιεχόμενά του.

Αφού βεβαιωθείτε ότι λειτουργεί σωστά η `selection_sort`, βάλτε την κλήση της σε σχόλιο, και καλέστε στο ίδιο σημείο τη `insertion_sort`. Βεβαιωθείτε ότι και αυτή ταξινομεί σωστά τον πίνακα. Μη σβήσετε το σχόλιο.

Στάδιο 3: Αναζήτηση

Γράψτε κώδικα στη `main` που εκτυπώνει χαρακτήρα αλλαγής γραμμής, 20 παύλες και πάλι χαρακτήρα αλλαγής γραμμής.

Γράψτε μια συνάρτηση με όνομα `find_name_aem` η οποία παίρνει ως παράμετρο μια συμβολοσειρά που θα περιέχει ονοματεπώνυμο, έναν ακέραιο που αναπαριστά αριθμό μητρώου κι ότι άλλο χρειάζεται και χρησιμοποιεί δυαδική αναζήτηση (`binary search`) για να βρει αν υπάρχει στον πίνακα φοιτητής με αυτό το όνομα και αριθμό μητρώου. Αν τον βρει, επιστρέφει τη θέση του στον πίνακα, διαφορετικά `-1`.

Γράψτε κώδικα στη `main` που

- Εκτυπώνει το μήνυμα "Enter name: "
- Διαβάζει ένα ονοματεπώνυμο. Θεωρήστε ότι θα δοθεί με την ίδια μορφή που είναι αποθηκευμένο (επώνυμο, κόμμα, μικρό).
- Εκτυπώνει το μήνυμα "Enter ID: "
- Διαβάζει ένα αριθμό μητρώου.
- Εκτυπώνει χαρακτήρα αλλαγής γραμμής και μετά καλεί τη συνάρτηση `find_name_aem` για να βρει φοιτητή με αυτό το όνομα και αριθμό μητρώου. Αν υπάρχει, εκτυπώνει το βαθμό του με ένα δεκαδικό ψηφίο και χαρακτήρα αλλαγής γραμμής. Αν δεν υπάρχει, εκτυπώνει το μήνυμα "N, I Not found" όπου N το ονοματεπώνυμο και I ο αριθμός μητρώου, και χαρακτήρα αλλαγής γραμμής.

Γράψτε μια συνάρτηση με όνομα `find_all_records` η οποία παίρνει ως παράμετρο μια συμβολοσειρά που θα περιέχει ονοματεπώνυμο κι ότι άλλο χρειάζεται και ψάχνει στον πίνακα όλους τους φοιτητές με αυτό το ονοματεπώνυμο. Για κάθε φοιτητή που βρίσκει, εκτυπώνει τον αριθμό μητρώου του, κόμμα, ένα κενό, το βαθμό του με ένα δεκαδικό ψηφίο και χαρακτήρα αλλαγής γραμμής.

Γράψτε κώδικα στη main που

- Εκτυπώνει το μήνυμα "Enter name: "
- Διαβάζει ένα ονοματεπώνυμο. Θεωρήστε ότι θα δοθεί με την ίδια μορφή που είναι αποθηκευμένο (επώνυμο, κόμμα, μικρό).
- Εκτυπώνει χαρακτήρα αλλαγής γραμμής και μετά καλεί τη συνάρτηση `find_all_records` για να βρει όλους τους φοιτητές με αυτό το ονοματεπώνυμο.
- Εκτυπώνει χαρακτήρα αλλαγής γραμμής.

Στάδιο 4: Ανάλυση αλγορίθμων ταξινόμησης

Αφού ολοκληρώσετε την άσκηση και λειτουργεί σωστά, θα προσθέσετε κώδικα στις υλοποιήσεις σας των αλγορίθμων ταξινόμησης που θα σας βοηθήσει να αναλύσετε την απόδοσή τους.

Για να αναλύσουμε την απόδοση ενός αλγορίθμου αναζήτησης μετράμε πόσες συγκρίσεις κάνει στην προσπάθειά του να ταξινομήσει τα δεδομένα.

Φτιάξτε στο Kate ένα απλό αρχείο κειμένου με όνομα `notesX_Y.txt` όπου X, Y τα AEM των δύο μελών της ομάδας. ΠΡΟΣΟΧΗ: Θέλουμε απλό αρχείο κειμένου, όχι doc κτλ.

Προσθέστε κατάλληλους μετρητές στις συναρτήσεις `insertion_sort` και `selection_sort` οι οποίοι μετρούν τον αριθμό συγκρίσεων της κάθε μίας. Εκτυπώστε αυτόν τον αριθμό στο τέλος της κάθε συνάρτησης.

Τρέξτε το πρόγραμμά σας για όλα τα ενδεικτικά αρχεία εισόδου χρησιμοποιώντας τον ένα αλγόριθμο και σημειώστε στο `notesX_Y.txt` τα αποτελέσματά σας: όνομα αλγορίθμου, όνομα αρχείου εισόδου, αριθμό συγκρίσεων.

Τρέξτε το πρόγραμμά σας για όλα τα ενδεικτικά αρχεία εισόδου χρησιμοποιώντας τον δεύτερο αλγόριθμο και σημειώστε στο `notesX_Y.txt` και τα νέα αποτελέσματα.

Θα δείτε κάποιες αποκλίσεις. Μελετήστε τα αρχεία εισόδου. Έχουν τα δεδομένα κάποια χαρακτηριστικά που κάνουν τον ένα αλγόριθμο καλύτερο ή χειρότερο από τον άλλο σε μερικές περιπτώσεις? **Γράψτε** τις παρατηρήσεις στο `notesX_Y.txt`.

Αφού ολοκληρώσετε την ανάλυσή σας, βάλτε σε σχόλια τον κώδικα που προσθέσατε σε αυτό το στάδιο για να κάνετε τις μετρήσεις σας (μετρητές, εκτύπωση αριθμού συγκρίσεων).

ΑΠΟΘΗΚΕΥΣΤΕ ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΑΣ ΣΕ ΑΡΧΕΙΟ ΜΕ ΟΝΟΜΑ `hw3b.c`

Χρήση fgets

Η εκφώνηση σας ζητά να χρησιμοποιήσετε τη συνάρτηση `fgets` για να διαβάσετε τα δεδομένα του προγράμματος. Το prototype της `fgets` βρίσκεται στο `stdio.h`.

Το πλεονέκτημα της `fgets` έναντι της `scanf` είναι ότι μας επιτρέπει να διαβάσουμε μία ολόκληρη γραμμή δεδομένων, ακόμη κι αν αυτή περιέχει κενά.

Η χρήση της για να διαβάσουμε από τη στάνταρ είσοδο (πληκτρολόγιο) είναι:

```
fgets(str, size, stdin);
```

Αυτή η κλήση θα έχει ως αποτέλεσμα να διαβαστούν από το πλήκτρολόγιο (`stdin`) το πολύ `size-1` χαρακτήρες και να αποθηκευτούν στον πίνακα χαρακτήρων `str`. Στην τελευταία θέση του πίνακα τοποθετείται ο χαρακτήρας `'\0'`.

Σημειώστε πως η `fgets` σταματάει να διαβάζει αν διαβαστούν `size-1` χαρακτήρες, ή βρει χαρακτήρα αλλαγής γραμμής (όποιο από τα δύο γίνει πρώτο). Αν βρει χαρακτήρα αλλαγής γραμμής (`'\n'`) πριν να έχουν διαβαστεί `size-1` χαρακτήρες, τότε θα αποθηκευτεί και αυτός στο `str`. Αν δεν θέλουμε να υπάρχει χαρακτήρας αλλαγής γραμμής στο τέλος του string, μπορούμε να τον εντοπίσουμε εύκολα με χρήση της συνάρτησης `strchr` ή της συνάρτησης `strlen` όπως δείξαμε στο φροντιστήριο, και να τον αλλάξουμε σε `'\0'`. Τα prototypes των `strchr`, `strlen` βρίσκονται στο `string.h`.

Για παράδειγμα, αν έχουμε

```
char str[5];
```

```
fgets(str, 5, stdin);
```

- αν γράψουμε στο πληκτρολόγιο `abc` και `enter`, το `str` θα περιέχει `"abc\n"`
- αν γράψουμε στο πληκτρολόγιο `abcd` και `enter`, το `str` θα περιέχει `"abcd"`
- αν γράψουμε στο πληκτρολόγιο `abcde` και `enter`, το `str` θα περιέχει `"abcde"`

Πώς να παραδώσετε τη δουλειά σας

Πριν παραδώσετε τα προγράμματά σας, προσθέστε σε σχόλια στην αρχή του κάθε αρχείου C τα πλήρη ονόματα και ΑΕΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με λατινικούς χαρακτήρες.

Κατασκευάστε ένα φάκελο με όνομα `hw3_ερwnumero1_AEM1_ερwnumero3_AEM2` και αντιγράψτε μέσα σε αυτόν τα `hw3a.c`, `hw3b.c`, `notesX_Y.txt`

Πηγαίνετε στο φάκελο μέσα στον οποίο βρίσκεται το `hw3_ερwnumero1_AEM1_ερwnumero2_AEM2` που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czvf hw3_ερwnumero1_AEM1_ερwnumero2_AEM2.tgz hw3_ερwnumero1_AEM1_ερwnumero2_AEM2
```

Στείλτε email:

- στη διεύθυνση ce120lab@gmail.com
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **CE120 hw3**
- και συνημμένο αρχείο το `hw3_ερwnumero1_AEM1_ερwnumero2_AEM2.tgz`

Επιβεβαιώστε ότι το άλλο μέλος της ομάδας έλαβε αντίγραφο και είναι σωστό.