

ΣΕΤ ΑΣΚΗΣΕΩΝ 2**ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2015-2016**

Προθεσμία: Τετάρτη 25/11/2015, 22:00

Διαβάστε πριν ξεκινήσετε

Διαβάστε την εκφώνηση προσεκτικά και “σχεδιάστε” το πρόγραμμά σας 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.

Μη διστάζετε να ζητήσετε βοήθεια! Χρησιμοποιήστε κατά προτίμηση το forum προγραμματισμού (<http://courses.inf.uth.gr/codingforums/>) ή, αν είναι απαραίτητο, email (π.χ. αν πραγματικά επιβάλλεται να στείλετε κάποιο κομμάτι κώδικα μαζί με το μήνυμά σας).

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.

Σας δίνουμε κάποια έτοιμα αρχεία και θα χρειαστεί να εγκαταστήσετε μια βιβλιοθήκη γραφικών στον υπολογιστή σας. Δείτε τις οδηγίες στο τέλος του φυλλαδίου.

Το μόνο αρχείο στο οποίο θα επέμβετε είναι το hw2 .c. Όπως θα δείτε περιλαμβάνει ήδη κάποιους ορισμούς που θα σας φανούν χρήσιμοι.

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
- Γενική μορφοποίηση προγράμματος (στοίχιση, ονόματα μεταβλητών, κτλ.)
- Σχεδιασμός προγράμματος και αποτελεσματική χρήση κατάλληλων δομών, μεταβλητών, σταθερών κτλ.
- 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)

Άσκηση 1: Εξάπλωση πυρκαγιάς

Θα γράψετε ένα πρόγραμμα το οποίο προσομοιώνει την εξάπλωση μιας πυρκαγιάς σε μια περιοχή. Για να διευκολυνθείτε στην απεικόνιση των αποτελεσμάτων θα σας παρέχουμε ένα γραφικό περιβάλλον στο οποίο θα προσαρμόσετε το πρόγραμμά σας.

Αναπαράσταση περιοχής

Η περιοχή αναπαρίσταται ως ένας διδιάστατος πίνακα ακεραίων με όνομα `grid` με `ROWS` γραμμές και `COLS` στήλες. Η ακέραια τιμή που αποθηκεύεται σε κάθε κελί του πίνακα κωδικοποιεί τα περιεχόμενά του. Έχουμε ήδη ορίσει τους παρακάτω τύπους περιεχομένων ως σταθερές:

- Φωτιά (`FIRE`, τιμή 0, κόκκινο χρώμα)
- Θάμνος (`BUSH`, τιμή 1, κίτρινο χρώμα)
- Δέντρο (`TREE`, τιμή 2, πράσινο χρώμα)
- Πέτρα (`ROCK`, τιμή 3, γκρι χρώμα)
- Καμμένη γη (`BURNT`, τιμή 4, μαύρο χρώμα)
- Κενό (`BLANK`, τιμή -1). Αυτό χρησιμοποιείται αποκλειστικά για την αρχικοποίηση του πίνακα και κατά πάσα πιθανότητα δε θα το χρειαστείτε.

Ο πίνακας `grid` είναι ήδη δηλωμένος στο `hw2.c` και τον αρχικοποιούμε εμείς με τυχαία αρχικά περιεχόμενα. Δε χρειάζεται να κάνετε κάτι παραπάνω για την αρχικοποίησή του.

Πέρασμα χρόνου

Όταν ξεκινά η εκτέλεση του προγράμματος, αρχικοποιείται ο πίνακας `grid` κι εμφανίζονται τα περιεχόμενά του σε ένα παράθυρο με κάθε κελί χρωματισμένο ανάλογα με το τι περιέχει. Το πρόγραμμα περιμένει μέχρι να κάνετε κλικ σε ένα οποιοδήποτε κελί με εύφλεκτα περιεχόμενα (δηλαδή δέντρο ή θάμνο) το οποίο θα είναι το πρώτο κελί που πιάνει φωτιά. Από εκεί και πέρα, κάθε φορά που πατάτε ένα πλήκτρο ή κάνετε κλικ οπουδήποτε (εκτός του `quit`) περνάει μία μονάδα χρόνου. Όλα αυτά γίνονται αυτόματα και δε χρειάζεται να κάνετε κάτι.

Κάθε φορά που περνά μια μονάδα χρόνου εκτελείται ο κώδικας στη συνάρτηση `step()` η οποία βρίσκεται στο αρχείο `hw2.c`. Ο κώδικας αυτός ανανεώνει τα περιεχόμενα του πίνακα με βάση τους κανόνες εξάπλωσης της πυρκαγιάς που περιγράφουμε πιο κάτω. **Εσείς θα πρέπει να γράψετε τον κώδικα που ανανεώνει τα περιεχόμενα του πίνακα.**

Στο τέλος κάθε βήματος το πρόγραμμα ξαναζωγραφίζει στο παράθυρο τον πίνακα με τα ανανεωμένα περιεχόμενα. Αυτό γίνεται αυτόματα και δε χρειάζεται να κάνετε κάτι.

Μπορείτε οποιαδήποτε στιγμή να τερματίσετε την εκτέλεση είτε κάνοντας κλικ στο κουμπί `QUIT` είτε πατώντας το πλήκτρο `Q` στο πληκτρολόγιο.

Κανόνες εξάπλωσης πυρκαγιάς

Τα περιεχόμενα ενός κελιού αλλάζουν με βάση τους παρακάτω κανόνες:

- Κάθε κελί επηρεάζεται μόνο από τα κελιά που βρίσκονται επάνω, κάτω, αριστερά ή δεξιά του (δηλαδή όχι διαγώνια).
- Φωτιά μπορούν να πιάσουν μόνο τα κελιά που περιέχουν δέντρα ή θάμνους.
- Αν ένα κελί έχει εύφλεκτα περιεχόμενα και κάποιο γειτονικό του έχει πιάσει φωτιά, τότε το κελί έχει μια πιθανότητα να πιάσει και αυτό φωτιά. Τα δέντρα έχουν `TREE_FIRE_PROBABILITY` να πιάσουν φωτιά αν ένας γείτονας καίγεται ενώ οι θάμνοι έχουν `BUSH_FIRE_PROBABILITY`. Τυπικά η πιθανότητα για ένα θάμνο είναι υψηλότερη από αυτή για ένα δέντρο.

- Ένα κελί που πιάνει φωτιά δεν θα περιέχει πια δέντρο ή θάμνο, αλλά φωτιά.
- Αν ένα δέντρο ή θάμνος έχει πιάσει φωτιά, τότε μετά από μερικά βήματα (`TREE_BURN_TIME` για το δέντρο και `BUSH_BURN_TIME` για το θάμνο) η φωτιά σβήνει και αντικαθίσταται από καμμένη γη.

Είναι πολύ σημαντικό να προσέξετε να μην αλλάξετε τα περιεχόμενα ενός κελιού πριν να έχει γίνει έλεγχος όλων των γειτονικών του. Το "τρικ" που χρησιμοποιούμε για να επιτύχουμε κάτι τέτοιο είναι να μαρκάρουμε όλες τις αλλαγές που πρέπει να έχουν ολοκληρωθεί στο τέλος του βήματος σε ένα βοηθητικό "παράλληλο" πίνακα, και μόνο στο τέλος του βήματος να τις μεταφέρουμε στον αρχικό μας. Το κελί (i, j) του βοηθητικού θα περιέχει τη νέα τιμή ή κάποια άλλη πληροφορία για το κελί (i, j) του `grid`.

Για παράδειγμα, αν ανιχνεύσετε ότι ένα κελί με δέντρο πρέπει να πιάσει φωτιά, το αφήνετε ως έχει μέχρι να τελειώσουν όλοι οι έλεγχοι αυτού του βήματος, αλλά μαρκάρετε το αντίστοιχο κελί του βοηθητικού. Στο τέλος του βήματος, ελέγχετε τον βοηθητικό πίνακα και μεταφέρετε τις πληροφορίες του στον καινούργιο.

Ένα άλλο σημείο που μπορείτε να χρησιμοποιήσετε βοηθητικό πίνακα είναι για να σημειώνετε πόσα βήματα έχει ακόμη ένα φλεγόμενο κελί μέχρι να μαρκαριστεί ως `BURNT`. Σε κάθε βήμα, η τιμή αυτή πρέπει να μειώνεται, κι όταν φτάσει στο μηδέν, τότε το κελί του πίνακα `grid` γίνεται `BURNT`.

Για να σας βοηθήσουμε, έχουμε ήδη ορίσει και αρχικοποιήσει δύο βοηθητικούς πίνακες στο `hw2.c`. Δείτε τα σχόλια στον κώδικα.

Τεχνικά ζητήματα

Προσπαθήσαμε να ξεχωρίσουμε όσο γίνεται το γραφικό κομμάτι από τον κώδικα που θα γράψετε. Ότι αλλαγές κάνετε πρέπει να γίνουν αποκλειστικά στο αρχείο `hw2.c`. Όποιον βοηθητικό πίνακα χρειαστείτε δηλώστε τον ως καθολική μεταβλητή. Οποιαδήποτε άλλη μεταβλητή χρειάζεστε (π.χ. i, j για προσπέλαση πινάκων) πρέπει να είναι δηλωμένη τοπικά μέσα στη συνάρτηση `step`.

Debugging/Βοήθεια

Μην προσπαθήσετε να τα κάνετε όλα μαζί. Σε πρώτη φάση υλοποιήστε την εξάπλωση της φωτιάς θεωρώντας ότι τα δέντρα και οι θάμνοι έχουν πιθανότητα 1 να πιάσουν φωτιά, και υποθέτοντας ότι δε σβήνει ποτέ.

Όταν λειτουργήσει σωστά αυτό, τότε αλλάξτε ότι χρειάζεται ώστε να λειτουργεί με διαφορετικές πιθανότητες.

Τέλος, προσθέστε ότι χρειάζεται ώστε να σβήνει η φωτιά μετά από μερικά βήματα.

Λόγω του γραφικού κομματιού ίσως δυσκολευτείτε στο debugging του κώδικά σας. Προτείνουμε τις παρακάτω τεχνικές:

- Δώστε την τιμή 0 στη σταθερά `IS_RANDOM`. Αυτό θα έχει ως αποτέλεσμα ο πίνακας `grid` να αρχικοποιείται πάντα με τον ίδιο τρόπο κι έτσι να μπορείτε να έχετε την ίδια συμπεριφορά προγράμματος σε κάθε εκτέλεση.
- Εισάγετε εντολές `printf` σε σημεία που θέλετε να ελέγξετε τι τιμές έχουν διάφορες μεταβλητές σας.
- Δώστε μικρότερες τιμές στις σταθερές `ROWS`, `COLS` που ελέγχουν το μέγεθος του πίνακα

Όπως πάντα, δώστε μεγάλη προσοχή στη στοίχιση. Είναι πολύ εύκολο να γίνει λάθος στο πού κλείνει κάποιο άγκιστρο αν δεν προσέξετε.

Δώστε καλά ονόματα στις μεταβλητές σας. Επαναχρησιμοποιήστε τους μετρητές των `for loops` όπου ενδείκνυται αντί να δηλώνετε καινούργιους.

Κάθε φορά που υλοποιείτε ένα κομμάτι, γράψτε σχόλια που να εξηγούν τον κώδικά σας. Θα βοηθήσουν και εσάς, και εμάς στη διόρθωση.

Χρησιμοποιήστε τις σταθερές που σας δίνουμε και για να είναι πιο ευανάγνωστο το πρόγραμμά σας αλλά και για να μπορείτε να κάνετε εύκολα ελέγχους με διαφορετικά μεγέθη.

Αποθηκεύστε το πρόγραμμά σας στο αρχείο με όνομα `hw2.c`

Παράρτημα: Γραφική βιβλιοθήκη και μεταγλώττιση προγράμματος

Για να μπορέσετε να μεταγλωττίσετε το πρόγραμμά σας θα χρειαστεί να έχετε εγκαταστήσει OpenGL και GLUT. Ακολουθήστε τις παρακάτω οδηγίες ανάλογα με το σύστημά σας. Για οποιοδήποτε πρόβλημα κάντε ερώτηση στο φόρουμ του μαθήματος.

Οδηγίες για MAC OS X

Software

Εφόσον έχετε XCode δε χρειάζεται να εγκαταστήσετε κάτι άλλο.

Μεταγλώττιση

Η εντολή για τη μεταγλώττιση είναι (σε μία γραμμή) :

```
gcc -Wall -Wno-deprecated-declarations -framework glut -framework  
OpenGL -framework Cocoa introGlutLib.c simulation.c -o hw2
```

Παρατηρήστε πως δεν αναφέρεται πουθενά το hw2.c. Μην σας απασχολεί αυτό - δεν υπάρχει λάθος.

Εναλλακτικά, εφόσον έχετε στον ίδιο φάκελο το αρχείο με όνομα Makefile, μπορείτε να γράψετε στο τερματικό σας

```
make osx
```

και θα εκτελεστεί αυτόματα η παραπάνω εντολή μεταγλώττισης.

Οδηγίες για Linux

Software

Βεβαιωθείτε ότι είστε συνδεδεμένοι στο internet.

Από το μενού του λειτουργικού σας, βρείτε το εργαλείο εγκατάστασης νέων προγραμμάτων (Install/Remove Software στο openSuse, Ubuntu Software Center στο Ubuntu.) Επιλέξτε το.

Στο παράθυρο που θα εμφανιστεί, θα υπάρχει ένα πεδίο αναζήτησης. Γράψτε `freeglut` σε αυτό. Ένα από τα αποτελέσματα είναι το `freeglut-devel` ή `freeglut3-dev`. Επιλέξτε το και πατήστε το κουμπί Install ή Accept για να το εγκαταστήσετε.

Μεταγλώττιση

Η εντολή για τη μεταγλώττιση είναι (σε μία γραμμή):

```
gcc -Wall introGlutLib.c simulation.c -o hw2 -lglut -lGLU -lGL -lm
```

Παρατηρήστε πως δεν αναφέρεται πουθενά το hw2.c. Μην σας απασχολεί αυτό - δεν υπάρχει λάθος.

Το `l` στα `-lglut`, `-lGLU` κτλ. είναι το αγγλικό γράμμα `l` (el) και όχι το ψηφίο `1` (ένα)

Εναλλακτικά, εφόσον έχετε στον ίδιο φάκελο το αρχείο με όνομα Makefile, μπορείτε να γράψετε στο τερματικό σας

```
make linux
```

και θα εκτελεστεί αυτόματα η παραπάνω εντολή μεταγλώττισης.

Άσκηση 2: Αποκρυπτογράφηση κειμένου

Εισαγωγή

Ένας εύκολος αλγόριθμος κρυπτογράφησης αντικαθιστά κάθε εκτυπώσιμο χαρακτήρα με αυτόν που έπεται κατά κάποιους χαρακτήρες στον πίνακα ASCII. Οι εκτυπώσιμοι χαρακτήρες ASCII ξεκινούν από το ' ' (κωδικός 32) και φτάνουν ως το '~' (κωδικός 126). Σε περίπτωση που περάσουμε τον τελευταίο εκτυπώσιμο χαρακτήρα, συνεχίζουμε κυκλικά από τον πρώτο.

Το κλειδί της κρυπτογράφησης είναι η απόσταση του χαρακτήρα που αντικαθίσταται από αυτόν που τον αντικαθιστά. Έτσι, για παράδειγμα, αν το κλειδί είναι 5, η λέξη wizard κρυπτογραφείται ως |n fwi. Τυχόν μη εκτυπώσιμοι χαρακτήρες (π.χ. '\n') παραμένουν ως έχουν.

Η βασική παρατήρηση για την αποκρυπτογράφηση είναι ότι ο πιο συχνός χαρακτήρας σε ένα κείμενο είναι κατά κανόνα το ' ' (κενό). Άρα, αν κανείς αναλύσει τη συχνότητα εμφάνισης των χαρακτήρων στο κρυπτογραφημένο κείμενο μπορεί βάσιμα να υποθέσει ότι ο χαρακτήρας με τη μεγαλύτερη συχνότητα εμφάνισης αντιστοιχεί στο αρχικό κείμενο στο κενό. Υπολογίζοντας την απόσταση αυτού του χαρακτήρα από το ' ' προκύπτει το κλειδί. Το κλειδί χρησιμοποιείται για την αποκρυπτογράφηση όλου του κειμένου, μετακινώντας κάθε χαρακτήρα πίσω (κυκλικά) τόσες θέσεις όσο η τιμή του κλειδιού.

Το πρόγραμμά σας θα πρέπει να διαβάζει από την είσοδο (με ανακατεύθυνση από αρχείο κειμένου) το κρυπτογραφημένο κείμενο, το οποίο μπορεί να έχει μήκος έως και 1024 χαρακτήρες. Αν το κείμενο περιέχει παραπάνω χαρακτήρες, αυτοί αγνοούνται. Το πρόγραμμα αναλύει τη συχνότητα εμφάνισης των χαρακτήρων στο κρυπτογραφημένο κείμενο (μετράει δηλαδή πόσες φορές εμφανίζεται κάθε εκτυπώσιμος χαρακτήρας), υπολογίζει το κλειδί και κατόπιν αποκρυπτογραφεί το κείμενο.

Η έξοδος του προγράμματος είναι το μήνυμα "KEY: " ακολουθούμενο από το κλειδί και 2 χαρακτήρες αλλαγής γραμμής, και μετά το μήνυμα "TEXT: " ακολουθούμενο από το αποκρυπτογραφημένο κείμενο και 1 χαρακτήρα αλλαγής γραμμής.

Ακολουθούν λεπτομερείς οδηγίες για το πώς πρέπει να λειτουργεί το πρόγραμμά σας και στάδια κατασκευής του. ΜΗΝ προσπαθήσετε να γράψετε όλο το πρόγραμμα σε ένα βήμα γιατί θα κάνετε λάθη και θα σας πάρει πολύ περισσότερο χρόνο.

Αποθηκεύστε το πρόγραμμα σε αρχείο με το όνομα `decoder.c` . Χρησιμοποιήστε σαν είσοδο (με ανακατεύθυνση) το κωδικοποιημένο αρχείο κειμένου `encoded.txt`.

Στάδιο 1: Εισαγωγή και αρχική επεξεργασία δεδομένων

Διαβάστε χαρακτήρα-προς-χαρακτήρα (`getchar`) από το πληκτρολόγιο ένα κείμενο έως ότου συναντήσετε τον ειδικό χαρακτήρα EOF ή έχετε διαβάσει το ανώτατο όριο χαρακτήρων που επιτρέπεται να περιέχει το κείμενο (όποιο από τα δύο συμβεί πρώτο).

Το κείμενο πρέπει να αποθηκευτεί σε κατάλληλο πίνακα.

Ταυτόχρονα, για κάθε εκτυπώσιμο χαρακτήρα μετρήστε το πλήθος εμφανίσεων αυτού στο κείμενο και αποθηκεύστε αυτή την πληροφορία σε πίνακα τόσων θέσεων όσοι και οι εκτυπώσιμοι χαρακτήρες ASCII.

Προσωρινός κώδικας: Εκτυπώστε τα περιεχόμενα και των δύο πινάκων και επιβεβαιώστε ότι αυτό το στάδιο λειτουργεί σωστά.

Το πρόγραμμά σας πρέπει να είναι γραμμένο με τέτοιο τρόπο ώστε αν κάποια στιγμή αυξηθεί το μέγιστο επιτρεπτό μέγεθος κειμένου, να μπορούν να γίνουν εύκολα και γρήγορα οι κατάλληλες αλλαγές στο πρόγραμμα.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 2: Εντοπισμός κενού ' '

Αφαιρέστε τον προσωρινό κώδικα που γράψατε στο στάδιο 1.

Εντοπίστε το χαρακτήρα με τη μεγαλύτερη συχνότητα (πλήθος εμφανίσεων) διατρέχοντας τον πίνακα στον οποίο είναι αποθηκευμένο το πλήθος εμφανίσεων για κάθε εκτυπώσιμο χαρακτήρα.

Προσωρινός κώδικας: Εκτυπώστε το χαρακτήρα (ή τη θέση) που βρήκατε και επιβεβαιώστε ότι είναι σωστός.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 3: Αποκωδικοποίηση κειμένου

Αφαιρέστε τον προσωρινό κώδικα που γράψατε στο στάδιο 2.

Χρησιμοποιήστε τη μέθοδο που περιγράφεται στην εισαγωγή για να υπολογίσετε το κλειδί, κι εκτυπώστε το στην οθόνη σε μήνυμα της μορφής :

KEY: k

όπου k το κλειδί που υπολογίσατε. Εκτυπώστε δύο χαρακτήρες αλλαγής γραμμής μετά το μήνυμα.

Έχοντας υπολογίσει το κλειδί, χρησιμοποιήστε τη μέθοδο που περιγράφεται στην εισαγωγή για να αποκωδικοποιήσετε το κείμενο. Εκτυπώστε το αποκωδικοποιημένο κείμενο στην οθόνη σε μήνυμα της μορφής:

TEXT: t

όπου t το αποκωδικοποιημένο κείμενο. Εκτυπώστε ένα χαρακτήρα αλλαγής γραμμής μετά το μήνυμα.

Βοήθεια:

Χρησιμοποιήστε τον τελεστή % για να κάνετε "κυκλική" διάτρεξη των εκτυπώσιμων χαρακτήρων, αλλά λάβετε υπόψη ότι δεν ξεκινάμε από το 0, αλλά από το χαρακτήρα με ASCII 32. Θα πρέπει να μετατρέψετε το σύνολο χαρακτήρων με ASCII κωδικούς από 32 έως και 126 σε σύνολο που ξεκινά από το 0 πριν μπορέσετε να κάνετε κυκλική διάτρεξη, και μετά να το ξαναμετατρέψετε στη σωστή κλίμακα. Δώστε ιδιαίτερη προσοχή ώστε να μην προκύψει ποτέ αρνητικός αριθμός κατά τη διάρκεια των πράξεών σας.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 4: Τελικός έλεγχος ορθότητας

Το πρόγραμμά σας πρέπει να λειτουργεί σωστά και να εκτυπώνει όλα τα μηνύματα και αποτελέσματα με τον τρόπο που σας περιγράφουμε. Για να μπορέσετε να ελέγξετε την ορθότητα θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου. Υπάρχει ένας εύκολος τρόπος να συγκρίνετε τα δικά σας αποτελέσματα με τα δικά μας:

Ας υποθέσουμε ότι το εκτελέσιμο πρόγραμμά σας λέγεται `decoder`, το ενδεικτικό αρχείο εισόδου λέγεται `encoded.txt` και το αντίστοιχο αρχείο εξόδου που σας έχουμε δώσει λέγεται `decoded.txt`

Η εντολή: `./decoder < encoded.txt > myoutput.txt`

εκτελεί το πρόγραμμά σας με την ενδεικτική είσοδο `input1.txt` και αποθηκεύει τα αποτελέσματα στο αρχείο εξόδου `myout1.txt`

Η εντολή `diff -b myoutput.txt decoded.txt`

συγκρίνει το δικό σας αρχείο εξόδου με το δικό μας. Αν υπάρχουν διαφορές, τις εμφανίζει (γραμμή-γραμμή). Αν δεν υπάρχουν διαφορές, δεν κάνει τίποτα.

Πρέπει το πρόγραμμα που θα μας παραδώσετε να παράγει έξοδο που δεν έχει διαφορές από τη δική μας.

Προσθέστε σε σχόλια στην αρχή του αρχείου τα πλήρη ονόματα και ΑΜ των μελών της ομάδας.

Αρχείο προς παράδοση: `decoder.c`

Πώς να παραδώσετε τη δουλειά σας

Πριν παραδώσετε το πρόγραμμά σας, προσθέστε σε σχόλια στην αρχή του αρχείου τα πλήρη ονόματα και ΑΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με λατινικούς χαρακτήρες.

Κατασκευάστε ένα φάκελο με όνομα `hw2_epwnumero1_AM1_epwnumero2_AM2` και αντιγράψτε μέσα σε αυτόν το `hw2.c` (θέλουμε ΜΟΝΟ το `hw2.c` για την άσκηση 1) και το `decoder.c`

Πηγαίνετε στο φάκελο μέσα στον οποίο βρίσκεται το `hw2_epwnumero1_AM1_epwnumero2_AM2` που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czvf hw2_epwnumero1_AM1_epwnumero2_AM2.tar.gz hw2_epwnumero1_AM1_epwnumero2_AM2
```

Στείλτε email:

- στη διεύθυνση **`ce120lab@gmail.com`**
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **`CE120 hw2`**
- και συνημμένο αρχείο το `hw2_epwnumero1_AM1_epwnumero2_AM2.tar.gz`