

ΣΕΤ ΑΣΚΗΣΕΩΝ 1**ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012**

Προθεσμία: 6/11/11, 22:00

Περιεχόμενα

- [Διαβάστε πριν ξεκινήσετε](#)
- [Εκφώνηση άσκησης](#)
- [Οδηγίες αποστολής άσκησης](#)
- [Παράρτημα Α: Ανακατεύθυνση](#)
- [Παράρτημα Β: Σύγκριση αρχείων](#)
- [Παράρτημα Γ: Μαθηματική βιβλιοθήκη](#)

Πριν ξεκινήσετε (ΔΙΑΒΑΣΤΕ ΑΥΤΗ ΤΗΝ ΕΝΟΤΗΤΑ!!)

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

Για κάθε στάδιο, αποφασίστε τι μεταβλητές θα χρειαστείτε, τι ονόματα θα τους δώσετε, αν χρειάζονται σταθερές κι αν ναι για ποιες ποσότητες, και τι δομές θα χρησιμοποιήσετε για κάθε λειτουργία.

Μη διστάζετε να ζητήσετε βοήθεια! Μπορείτε να χρησιμοποιήσετε το forum προγραμματισμού (<http://inf-server.inf.uth.gr/courses/coding/>) και φυσικά email.

Μην προσπαθήσετε να γράψετε το πρόγραμμα μια κι έξω. Για κάθε στάδιο που ολοκληρώνετε, βεβαιωθείτε ότι κάνει compile και λειτουργεί σωστά και μετά να προχωράτε στο επόμενο στάδιο. Κάθε φορά που τελειώνετε ένα στάδιο, να αποθηκεύετε ένα αντίγραφο του προγράμματος όπως είναι μέχρι εκείνο το σημείο. Έτσι, αν σε κάποιο επόμενο στάδιο καταστραφεί το αρχείο σας, θα έχετε μια παλιότερη έκδοση από την οποία μπορείτε να συνεχίσετε. Τις επιμέρους "εκδόσεις" να τις ονομάζετε με διαφορετικά ονόματα (πχ. hw1stadio1.c, hw1stadio2.c)

Σε κάθε στάδιο σας δίνουμε μια εκτίμηση της δυσκολίας του (διαβάθμιση απο 1 έως 4).

Η εργασία αυτή μπορεί να γίνει σε ομάδες μέχρι 2 ατόμων. Μπορείτε να συζητάτε τις ασκήσεις με συμμαθητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιοδήποτε τρόπο.

Ξεκινήστε νωρίς! Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ό,τι περιμένετε.

Εκπρόθεσμες ασκήσεις δε γίνονται δεκτές.

Άσκηση: Σταθμός διοδίων

Εισαγωγή

Θα γράψετε ένα πρόγραμμα το οποίο χρησιμοποιείται από ένα πλήρως αυτοματοποιημένο σταθμό διοδίων για να διαχειριστεί τις πληρωμές. Το πρόγραμμα ανιχνεύει τον τύπο κάθε οχήματος που περνά, αποφασίζει το αντίτιμο, δέχεται το ποσό, υπολογίζει αν είναι αρκετό και αν χρειάζονται ρέστα, υπολογίζει και δίνει στον οδηγό τα ρέστα και στο τέλος της ημέρας καταγράφει στατιστικά στοιχεία για τη λειτουργία του σταθμού.

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.

Στάδιο 1: Ανίχνευση τύπου οχήματος (★)

Γράψτε ένα πρόγραμμα το οποίο ζητά τον τύπο του οχήματος ("Enter vehicle type: "), διαβάζει από το πληκτρολόγιο τον τύπο, και εκτυπώνει στην οθόνη το μήνυμα "PAY xx.xx" όπου xx.xx είναι το ποσό που πρέπει να πληρωθεί από τον οδηγό για το συγκεκριμένο τύπο οχήματος. Στην επόμενη γραμμή εκτυπώνει το χαρακτήρα > ακολουθούμενο από ένα κενό.

Ο τύπος δίνεται ως χαρακτήρας και μπορεί να είναι ένας από τους παρακάτω:

- M ή m για μοτοσικλέτα
- C ή c για αυτοκίνητο
- B ή b για λεωφορείο
- T ή t για φορτηγό

Το ποσό είναι πάντα το ίδιο για κάθε συγκεκριμένο τύπο και είναι :

- €1.25 για μοτοσικλέτες
- €2.35 για αυτοκίνητα
- €8.40 για φορτηγά
- €16.50 για λεωφορεία

Το πρόγραμμα πρέπει να εκτυπώνει ακριβώς 2 δεκαδικά ψηφία για το οφειλούμενο ποσό.

Το πρόγραμμά σας πρέπει να είναι γραμμένο με τέτοιο τρόπο ώστε αν κάποια στιγμή αυξηθούν οι τιμές, να είναι εύκολο ένας προγραμματιστής να κάνει τις κατάλληλες αλλαγές στο πρόγραμμα.

Αποθηκεύστε το πρόγραμμά σας σε ένα αρχείο με όνομα hw1.c

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

Στάδιο 2: Ανίχνευση πολλαπλών οχημάτων(★ ★)

Μεταβάλετε τη δομή επανάληψης στο πρόγραμμά σας ώστε να μπορεί να επαναλαμβάνει την παραπάνω διαδικασία όσο έρχονται οχήματα. Θεωρείστε ότι τα οχήματα σταματούν να έρχονται αν ως τύπος δοθεί ο χαρακτήρας '0' (μηδέν).

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

Στάδιο 3: Εισαγωγή πληρωμής(★ ★ ★)

Το σύμβολο > (το οποίο εκτυπώσατε στο στάδιο 1) προτρέπει τον οδηγό να πληρώσει το ποσό που οφείλει. Προσθέστε κώδικα στο πρόγραμμά σας ώστε μετά την εκτύπωση του > να διαβάζει από το πληκτρολόγιο το ποσό που δίνει ο οδηγός.

Αν το ποσό δεν είναι αρκετό, το πρόγραμμα πρέπει να εκτυπώνει το μήνυμα "PAY xx.xx more" όπου xx.xx είναι το επιπλέον ποσό που χρειάζεται να πληρώσει ο οδηγός. Στην επόμενη γραμμή εκτυπώνεται και πάλι το σύμβολο > και ένα κενό, και το πρόγραμμα πρέπει να ξαναδιαβάσει το επιπλέον ποσό που θα πληρώσει ο οδηγός.

Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου ο οδηγός να πληρώσει τουλάχιστον όσο χρωστάει (μπορεί να δώσει παραπάνω χρήματα).

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

Στάδιο 4: Υπολογισμός ρέστων(★ ★ ★ ★)

Οι οδηγοί δεν έχουν πάντα το ακριβές ποσό και το σύστημα πρέπει να έχει τη δυνατότητα να υπολογίσει και να επιστρέψει τα ρέστα.

Στο προηγούμενο στάδιο, το πρόγραμμα σταματά να ζητάει χρήματα όταν ο οδηγός έχει δώσει είτε ακριβώς το ποσό που πρέπει, είτε περισσότερο.

Προσθέστε κώδικα στο πρόγραμμά σας που ελέγχει αν το ποσό που έδωσε ο οδηγός είναι μεγαλύτερο από το οφειλούμενο και, αν ναι, υπολογίζει πόσα ρέστα πρέπει να του επιστρέψει. Το πρόγραμμά σας πρέπει να μπορεί να υπολογίσει ακριβώς πόσα νομίσματα κάθε τύπου να επιστρέψει.

Μπορείτε να υποθέσετε πως τα νομίσματα που μπορούν να δοθούν ως ρέστα είναι των 20 ευρώ, 5 ευρώ, 1 ευρώ, 10 λεπτών και ενός λεπτού. Για παράδειγμα, αν ένας οδηγός φορτηγού δώσει 20 ευρώ, τότε τα ρέστα του είναι δύο χαρτονομίσματα των 5 ευρώ, 1 νόμισμα του 1 ευρώ, και 6 νομίσματα των 10 λεπτών. Τα ρέστα πρέπει να εκτυπώνονται με σειρά από μεγαλύτερη προς μικρότερη αξία και με τον τρόπο που φαίνεται στο παρακάτω παράδειγμα:

Amount Received: 20.00 euro

Change:

2 of 5 euro

1 of 1 euro

6 of 10 cents

Μπορείτε να υποθέσετε ότι το σύστημα έχει πάντα τα νομίσματα που χρειάζεται. Αν ο οδηγός πληρώσει το ακριβές ποσό, τότε απλά εκτυπώνεται:

Change: 0

Αν τα ρέστα δεν περιλαμβάνουν κάποιο τύπο νομίσματος, τότε αυτός δεν εκτυπώνεται. Για παράδειγμα, αν τα ρέστα είναι 10 ευρώ και 10 λεπτά, τότε το πρόγραμμα δεν πρέπει να εκτυπώσει ότι θα δοθούν μηδέν νομίσματα του ενός ευρώ.

Στάδιο 5: Υπολογισμός στατιστικών (★ ★ ★)

Προσθέστε κώδικα στο πρόγραμμά σας, ώστε να μετρά και να εκτυπώνει πόσα οχήματα συνολικά έχουν περάσει από το σταθμό μέσα σε μια μέρα, πόσα είναι τα έσοδα ανά τύπο οχήματος και πόσα είναι τα συνολικά έσοδα. Τα στατιστικά εκτυπώνονται αφού έχουν περάσει όλα τα οχήματα, και με την παρακάτω μορφή:

```

Number of vehicles:    uuuuuuuu
Motorcycles:          vvvv.vv
Cars:                 www.w
Buses:                xxxx.xx
Trucks:               yyyy.yy
Total:                zzzzz.z

```

Κάθε αριθμός εκτός του Total καταλαμβάνει 7 θέσεις. Οι αριθμοί κινητής υποδιαστολής έχουν 2 δεκαδικά ψηφία. Το Total καταλαμβάνει 8 θέσεις με 2 δεκαδικά ψηφία.

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

Στάδιο 6: 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 `hw1`, το ενδεικτικό αρχείο εισόδου λέγεται `input1.txt` και το αντίστοιχο αρχείο εξόδου που σας έχουμε δώσει λέγεται `stdout1.txt`

Η εντολή:

```
./hw1 < input1.txt > myout1.txt
```

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

Η εντολή

```
diff -b myout1.txt stdout1.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.

Για περισσότερες πληροφορίες δείτε το σχετικό παράρτημα στο τέλος του φυλλαδίου.

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

Υπάρχουν κάποιες τιμές ελέγχου για τις οποίες το πρόγραμμα φαίνεται να παράγει λάθος αποτελέσματα κατά τον υπολογισμό των λεπτών, στα ρέστα. Για παράδειγμα, δοκιμάστε ένα αυτοκίνητο του οποίου ο οδηγός πληρώνει 20 ευρώ. Θα δείτε ότι υπολογίζει 4 αντί για 5 λεπτά στο τέλος. Αυτό οφείλεται σε περιορισμούς λόγω της αναπαράστασης αριθμών κινητής υποδιαστολής (0.4999999 αντί για 0.50). Το ίδιο πρόβλημα μπορεί να σας παρουσιαστεί όταν υπολογίζετε το επιπλέον ποσό που πρέπει να πληρώσει ο οδηγός. Μπορείτε να χρησιμοποιήσετε τις συναρτήσεις `round` και `lround` για να κάνετε κατάλληλες στρογγυλοποιήσεις ή να τις κάνετε με κάποιο άλλο έξυπνο τρόπο.

Αν χρησιμοποιήσετε τη `round` ή την `lround` πρέπει να κάνετε `include` το `math.h` και να κάνετε `compile` με `-lm`.

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.

Αρχεία προς παράδοση: hw1.c

Πώς να παραδώσετε τη δουλειά σας

(Ακολουθείστε τις οδηγίες ακριβώς αλλιώς μπορεί να μη δούμε τα αρχεία σας)

Κατασκευάστε ένα φάκελο με όνομα `erwnumο1_AM1_erwnumο2_AM2` και αντιγράψτε μέσα σε αυτόν το `hw1.c`

Πηγαίνετε στο φάκελο μέσα στον οποίο βρίσκεται το `erwnumο1_AM1_erwnumο2_AM2` που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czf erwnumο1_AM1_erwnumο2_AM2.tar.gz erwnumο1_AM1_erwnumο2_AM2
```

Στείλτε email:

- στη διεύθυνση **`ce120lab@gmail.com`**
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **CE120 hw1**
- και επικολλημένο αρχείο το `erwnumο1_AM1_erwnumο2_AM2.tar.gz`

Το πρόγραμμά σας θα βαθμολογηθεί ως προς:

- ορθότητα υπολογισμών
- σωστή επιλογή τύπων
- σωστή επιλογή και χρήση δομών ελέγχου κι επανάληψης
- αναγνωσιμότητα (ονόματα, στοίχιση, σχολιασμός, μορφή κώδικα)
- φορμαρισμένη είσοδος/έξοδος

Παράρτημα Α: Ανακατεύθυνση στο Linux

Μέχρι στιγμής, έχετε δει πως να γράφετε ένα πρόγραμμα που διαβάζει από το πληκτρολόγιο και γράφει στην οθόνη. Αυτό που συμβαίνει στην πραγματικότητα είναι ότι το πρόγραμμα διαβάζει από το λεγόμενο standard input (stdin, προκαθορισμένη είσοδος) και γράφει στο λεγόμενο standard output (stdout, προκαθορισμένη έξοδος). Τυπικά, το standard input είναι συνδεδεμένο με το πληκτρολόγιο και το standard output με την οθόνη.

Στην πρώτη άσκηση του σημερινού εργαστηρίου θέλουμε να διαβάσουμε από το πληκτρολόγιο 120 αριθμούς. Αυτό είναι αρκετά μεγάλο νούμερο - θα ήταν πιο πρακτικό αν είχαμε αυτούς τους αριθμούς σε ένα αρχείο και διαβάζαμε από το αρχείο. Το Linux μας επιτρέπει να επιτύχουμε κάτι τέτοιο με την "ανακατεύθυνση". Όταν το πρόγραμμα παράγει έξοδο, μπορούμε να την ανακατευθύνουμε από την οθόνη σε ένα αρχείο. Ομοίως, όταν το πρόγραμμα θέλει να διαβάσει είσοδο, την ανακατευθύνουμε ώστε να γίνεται από αρχείο, αντί από το πληκτρολόγιο.

Για να ανακατευθύνουμε την είσοδο χρησιμοποιούμε το ειδικό σύμβολο <

Παράδειγμα: `./myprogram < input`

Το πρόγραμμα myprogram παίρνει την είσοδό του από το αρχείο input αντί για το πληκτρολόγιο

Για να ανακατευθύνουμε την έξοδο χρησιμοποιούμε το ειδικό σύμβολο >

Παράδειγμα: `./myprogram > output`

Το πρόγραμμα myprogram γράφει την έξοδό του στο αρχείο output αντί για την οθόνη

Παράδειγμα: `./myprogram < input > output`

Το πρόγραμμα myprogram παίρνει την είσοδό του από το αρχείο input και γράφει την έξοδό του στο αρχείο output.

Γράψτε σε ένα αρχείο με όνομα `data.c` το παρακάτω πρόγραμμα:

```
#include<stdio.h>

int main(int argc, char *argv[]) {
    int i;
    for (i=0; i<120; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```

Κάντε compile κι εκτελέστε το. Θα δείτε να τυπώνει στην οθόνη 120 αριθμούς.

Τώρα, εκτελέστε το με ανακατεύθυνση εξόδου προς ένα αρχείο με όνομα `data`:

`./data > numbers.txt`

Δε θα εμφανιστεί τίποτα στην οθόνη, αλλά αν ανοίξετε το `numbers.txt` θα δείτε μέσα τους 120 ακεραίους.

Παράρτημα Β: Σύγκριση αρχείων

Η εντολή `diff` του Linux χρησιμοποιείται για να ελέγχουμε αν δυο αρχεία είναι ίδια. Δοκιμάστε τη! Φτιάξτε ένα αντίγραφο του αρχείου `data.c` από το προηγούμενο παράρτημα και ονομάστε το `data2.c`. Μετά, στη γραμμή εντολής γράψτε:

```
diff -b data.c data2.c
```

Δε θα βγεί τίποτα γιατί τα αρχεία είναι ίδια. Τώρα, κάντε κάποιες αλλαγές στο `data2.c` και ξαναδοκιμάστε. Το `diff` θα παρουσιάσει όλες τις διαφορές ανάμεσα στα δύο προγράμματα.

Μπορείτε να χρησιμοποιείτε ανακατεύθυνση για να εκτυπώνετε την έξοδο ενός προγράμματός σας σε αρχείο, να κάνετε το ίδιο και για την έξοδο του εκάστοτε εκτελέσιμου που σας δίνουμε και μετά να χρησιμοποιήσετε `diff` για να δείτε αν τα αποτελέσματα είναι ίδια. Με τον ίδιο τρόπο θα ελέγχουμε κι εμείς τα αποτελέσματα των προγραμμάτων σας όποτε είναι δυνατό.

Παράρτημα C: Μαθηματική βιβλιοθήκη

Αν το πρόγραμμά σας χρησιμοποιεί συναρτήσεις από τη μαθηματική βιβλιοθήκη (όπως οι `pow`, `floor`, `sin`, `round`, κτλ), τότε πρέπει να κάνετε δύο πράγματα για παραχθεί εκτελέσιμο:

1. Να προσθέσετε τη γραμμή `#include<math.h>` στην αρχή του προγράμματος.
2. Όταν κάνετε `compile` να προσθέσετε την επιλογή `-lm` στο τέλος της εντολής `gcc`