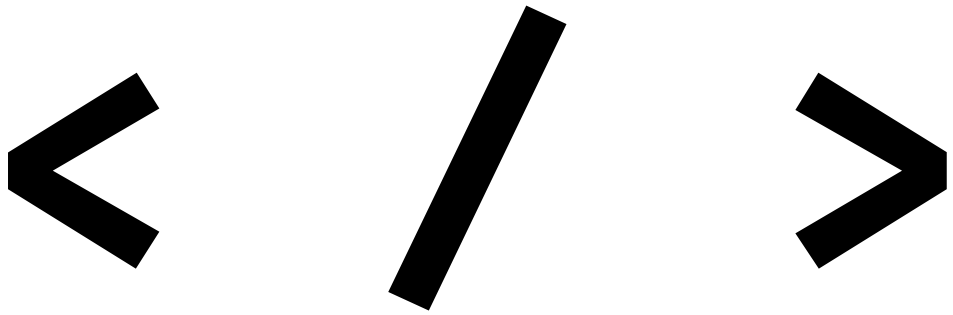


Πρώτη Εργασία

Επεξήγηση αρχείων κώδικα

Χρήστος Ζόια
Α.Μ. 1122929

Christos Joia
R.N. 1122929

Άσκηση 1

Ώστε το πρόγραμμά μας να δέχεται τρίγωνα οποιονδήποτε διαστάσεων, οι μεταβλητές που αποθηκεύουν τα μήκη των πλευρών του τριγώνου θα πρέπει να είναι πραγματικού τύπου, άρα τις δηλώνουμε με `double a, b, c;`

Το τρίγωνο πρέπει να έχει θετικά μήκη πλευρών, άρα κάνουμε επικύρωση των τιμών που δίνονται από τον χρήση χρησιμοποιώντας έναν βρόχο `do while`, ο οποίος επαναλαμβάνεται αν οποιαδήποτε από τις μεταβλητές a, b ή c είναι αρνητικές ή μηδέν. Αυτή η συνθήκη σε κώδικα C γράφεται ως εξής: `(a <= 0 || b <= 0 || c <= 0)`.

Όπως δίνεται από την εκφώνηση, για να είναι έγκυρο ένα τρίγωνο θα πρέπει να ισχύουν ταυτόχρονα οι τρεις ανισότητες `a + b > c`, `a + c > b` και `b + c > a`, δηλαδή η σύζευξη αυτών. Αυτή η συνθήκη σε κώδικα C γράφεται ως εξής: `((a + b > c) && (a + c > b) && (b + c > a))`. Αν ισχύει ή όχι αυτή η συνθήκη εκτυπώνουμε το αντίστοιχο μήνυμα στον χρήστη για την ύπαρξη ή όχι ενός τέτοιου τριγώνου.

Χρήστος Ζόια
A.M. 1122929

Christos Joia
R.N. 1122929

Άσκηση 2

Ο χρήστης εισάγει την ώρα σε μορφή hh:mm:ss, δηλώνουμε 3 ακέραιες μεταβλητές h, m, και s αντίστοιχα. Το εύρος των τιμών που μπορεί να πάρει η μεταβλητή h είναι το σύνολο ακεραίων [0, 23], ενώ οι μεταβλητές m και s παίρνουν τιμές στο σύνολο [0, 59], άρα αν οποιαδήποτε μεταβλητή είναι εκτός αυτών των δύο συνόλων ζητάμε από τον χρήστη να επανεισάγει την ώρα. Σε κώδικα C ο έλεγχος αρνητικών τιμών γράφεται `(h < 0 || min < 0 || sec < 0)`, ενώ ο έλεγχος ανώτατου ορίου γράφεται `(h >= 24 || min >= 60 || sec >= 60)`. Η επικύρωση των τιμών γίνεται με έναν βρόχο do while με διάζευξη αυτών των δύο συνθηκών, δηλαδή `((h >= 24 || min >= 60 || sec >= 60) || (h < 0 || min < 0 || sec < 0))`.

Για να δούμε πόση ώρα απομένει μέχρι τα μεσάνυχτα πρέπει να κάνουμε 24:00:00 - hh:mm:ss, όπου hh:mm:ss η δοσμένη ώρα. Ένας απλός τρόπος να γίνει αυτή η αφαίρεση είναι να μετατρέψουμε την ώρα της μορφής hh:mm:ss σε δευτερόλεπτα από την αρχή της ημέρας και να τα αφαιρέσουμε από τα δευτερόλεπτα που διαρκεί μία ημέρα, δηλαδή από 24•60•60.

Αρχικά πρέπει να μετατρέψουμε την ώρα της μορφής hh:mm:ss σε δευτερόλεπτα από την αρχή της ημέρας, άρα κάνουμε τις γνωστές μετατροπές (1m=60s, 1h=60m=60•60s), αθροίζουμε τα αποτελέσματα και τα αποθηκεύουμε σε μία μεταβλητή, το οποίο σε κώδικα C γράφεται ως εξής: `secFromMidnight = h * 60 * 60 + min * 60 + sec;`

Κάνοντας την αφαίρεση 24h - [δοσμένη ώρα] θα έχουμε την ώρα μέχρι τα μεσάνυχτα, άρα αφαιρούμε τα δευτερόλεπτα που πέρασαν από την αρχή της ημέρας από τα 24•60•60 δευτερόλεπτα και το αποθηκεύουμε σε μία μεταβλητή, το οποίο σε κώδικα C γράφεται `sec2Midnight = 24 * 60 * 60 – secFromMidnight;`

Έχοντας υπολογίσει τα δευτερόλεπτα που απομένουν μέχρι τα μεσάνυχτα, θα πρέπει να τα επιστρέψουμε στον χρήστη σε μορφή hh:mm:ss. Αρχικά υπολογίζουμε τις ώρες κάνοντας τη διαίρεση `sec2Midnight / (60 * 60)`. Το πηλίκο αυτής της διαίρεσης μας δίνει τις ώρες που απομένουν μέχρι τα μεσάνυχτα, ενώ το υπόλοιπο μας δίνει τα λεπτά και τα δευτερόλεπτα. Ομοίως υπολογίζουμε τα λεπτά: `sec2Midnight / 60`, με πηλίκο τα λεπτά και υπόλοιπο τα δευτερόλεπτα.

Έχοντας αποθηκεύσει τα αποτελέσματα των παραπάνω υπολογισμών σε μεταβλητές h, m, και s αντίστοιχα, τις εκτυπώνουμε στον χρήστη με δύο ψηφία ώστε ακόμα και μονοψήφιες τιμές να εκτυπώνεται το 0 μπροστά: `printf("%02d:%02d:%02d", h, min, sec);`

Άσκηση 3

Το πλήθος των μπουκαλιών είναι ακέραιος αριθμός, πιθανά πολύ μεγάλος, ενώ η τιμή είναι πραγματικός, άρα κάνουμε τη δήλωση τύπων: `long small, large; double total;`

Το πλήθος των μπουκαλιών πρέπει να είναι θετικό ή μηδέν, το οποίο ελέγχεται με βρόχους `do while`, οι οποίοι επαναλαμβάνονται αν το πλήθος είναι αρνητικό.

Για να υπολογίσουμε το σύνολο κάνουμε `total = small * 0.008 + large * 0.02;`

Η έκπτωση των 200€ περιέχεται και στην έκπτωση των 600€, άρα πρώτα ελέγχεται η συνθήκη των 600€ με σχήμα `if else`.

Για λόγους ταχύτητας, η 20% έκπτωση γίνεται 100%-20%=80%=0.8 της αρχικής τιμής και ομοίως η 8% έκπτωση γίνεται 0.92 της αρχικής, άρα κάνουμε `total *= 0.8;` και `total *= 0.92;` αντίστοιχα.

Η εκτύπωση της τιμής πρέπει να γίνει με δύο δεκαδικά ψηφία, άρα εκτυπώνεται με μορφοποίηση `%.2f`: `printf("Total cost: %.2f€\n", total);`

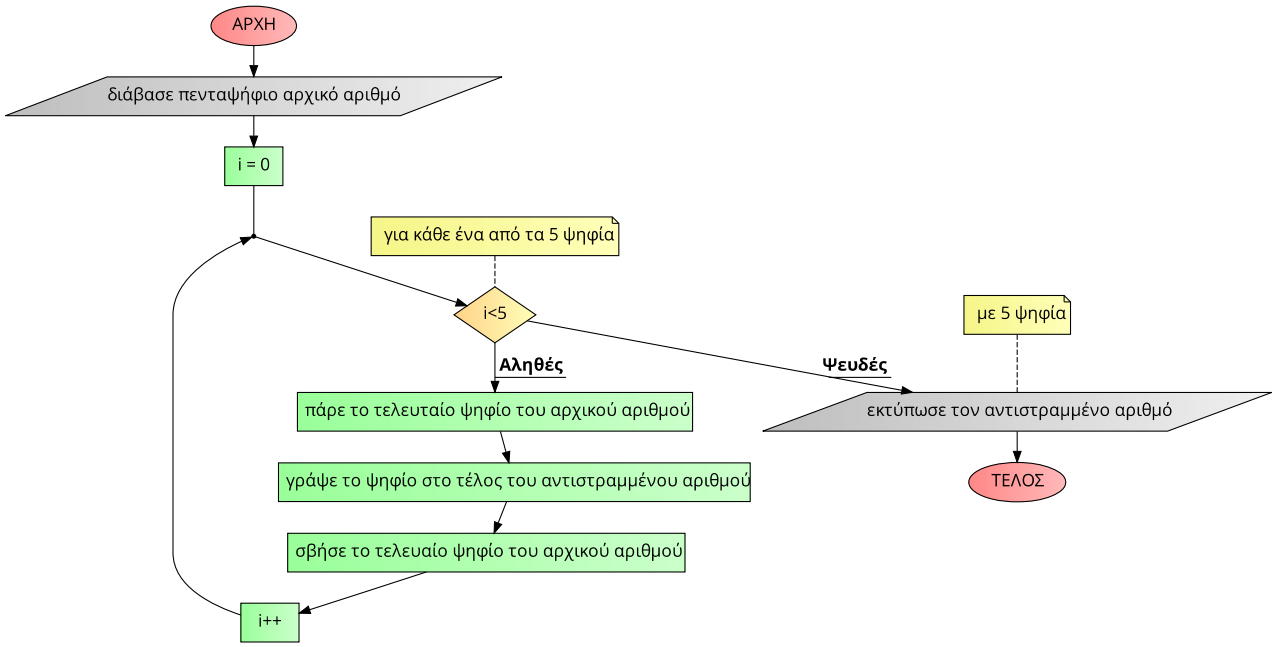
Χρήστος Ζόια
A.M. 1122929

Christos Joia
R.N. 1122929

Άσκηση 4

Όλοι οι πενταψήφιοι είναι μεταξύ 10^5 και 10^6, άρα η επικύρωση εισαγόμενων τιμών γίνεται με έναν βρόχο `do while (number < 10000 || number >= 100000);`

Για να αντιστρέψουμε τον αριθμό θα εκτελέσουμε τον παρακάτω αλγόριθμο:



Αυτός ο αλγόριθμος είναι ευνόητος για έναν άνθρωπο που τον εκτελεί με μολύβι και χαρτί, αλλά ώστε να μπορεί να εκτελεστεί από τον υπολογιστή χρειάζεται να είναι γραμμένος σε μαθηματικά.

Το βήμα 1 γίνεται απλά με την πράξη αριθμός mod 10.

Το βήμα 2 για να πραγματοποιηθεί αριθμητικά θα πρέπει να ενεργήσουμε ως εξής:

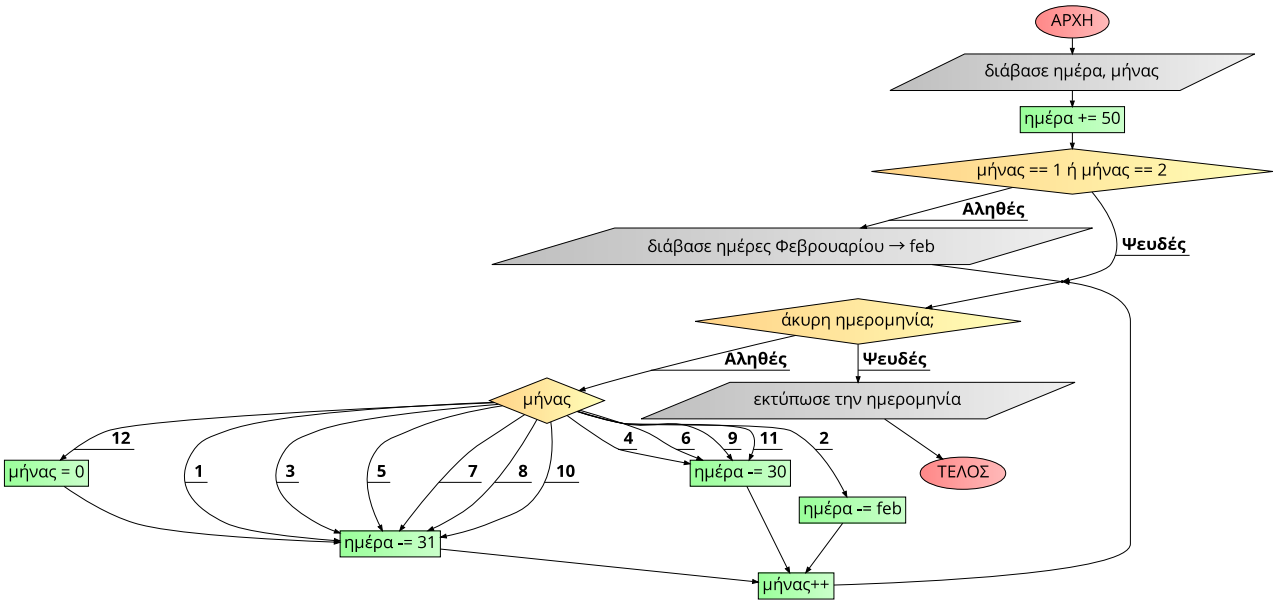
i. Ολισθαίνουμε τον αντιστραμμένο αριθμό προς τα αριστερά, το οποίο είναι αντίστοιχο με την πράξη αντιστραμμένος•10, έτσι ώστε να λήγει σε 0 (πχ. 12→120)

ii. Προσθέτουμε το τελευταίο ψηφίο του αρχικού αριθμού (πχ. το τελευταίο ψηφίο είναι 3, άρα 120+3→123)

Το βήμα 3 γίνεται απλά με την ακέραια διαίρεση του αρχικού αριθμού με το 10.

Τέλος, εκτυπώνουμε τον αριθμό με 5 ψηφία, ώστε ακόμα και αριθμοί που τελειώνουν σε μηδενικά να εμφανίζονται στην αρχή (πχ. 12500 → 00521).

Άσκηση 5



Θα ζητήσουμε από τον χρήστη τον αριθμό του μήνα, την ημέρα και τις ημέρες του Φεβρουαρίου, άρα θα χρειαστούμε 3 ακέραιες μεταβλητές: `int mon, feb, days;`

Αρχικά διαβάζουμε τον αριθμό του μήνα και τον επικυρώνουμε με έναν βρόχο `do while`, ο οποίος επαναλαμβάνεται αν η εισαγόμενη τιμή δεν ανήκει στο σύνολο [1, 12].

Έπειτα διαβάζουμε τον αριθμό των ημερών του μήνα που εισάχθηκαν. Διακρίνουμε 3 περιπτώσεις: ο μήνας να έχει 31 ημέρες, 30 ημέρες ή να είναι ο Φεβρουάριος, ο οποίος έχει 28 ή 29 ημέρες. Η εξέταση αυτών των περιπτώσεων γίνεται με δομή επιλογής `switch case`.

Ωστόσο, παρατηρούμε ότι 31+28-50=9, άρα οι ημέρες του Φεβρουαρίου πρέπει να εξετάζονται και σε περιπτώσεις όπου η ημερομηνία είναι μετά τις 9 Ιανουαρίου. Για λόγους απλότητας θα εξετάζονται για όλες τις ημερομηνίες του Ιανουαρίου.

Για να υπολογίσουμε την ημερομηνία 50 ημέρες μετά, αρχικά προσθέτουμε αφελώς 50 στη μεταβλητή `days` (πχ. 5 Μαρτίου → 55 Μαρτίου). Προφανώς, αυτό οδηγεί σε «υπερχείλιση» του αριθμού ημερών, η οποία πρέπει να διορθωθεί, οπότε με διαδοχικές αφαιρέσεις 28, 29, 30 ή 31 ημερών, ανάλογα με την περίπτωση, και μοναδιαία αύξηση του αριθμού του μήνα καταλήγουμε σε μία έγκυρη ημερομηνία. Στην περίπτωση του Δεκεμβρίου πρέπει να επαναφέρουμε τον αριθμό του μήνα σε 1.

Η διόρθωση αυτή γίνεται με έναν βρόχο ο οποίος επαναλαμβάνεται μέχρι να καταλήξουμε σε έγκυρη ημερομηνία. Ο έλεγχος γίνεται με σχήμα επιλογής `switch case`, όπου σε κάθε περίπτωση αν η ημερομηνία δεν είναι έγκυρη (αριθμός μερών πάνω από 28, 29, 30 ή 31 μέρες αντίστοιχα) γίνεται διόρθωση και επανάληψη του βρόχου.

Άσκηση 6

Θα χρειαστούμε έναν πίνακα μεγέθους 6 που περιέχει ακέραιες τιμές ο οποίος θα αποθηκεύει κάθε οκτάδα της διεύθυνσης MAC: `int oct[6];`

Η κάθε οκτάδα θα δίνεται σε δεκαεξαδική μορφή, άρα θα κάνουμε:
`scanf("%x:%x:%x:%x:%x:%x",&oct[0], &oct[1], &oct[2], &oct[3], &oct[4], &oct[5]);`

Η διεύθυνση MAC της οποίας όλες οι οκτάδες είναι `0xFF` είναι διεύθυνση broadcast, άρα ελέγχουμε αν όλα τα κελιά `oct[X]` είναι ίσα με 0xFF. Αν η πρώτη οκτάδα είναι άρτια πρόκειται για διεύθυνση unicast, αλλιώς πρόκειται για διεύθυνση multicast. Η αρτιότητα της `oct[0]` ελέγχεται με την πράξη `oct[0] mod 2`, η οποία όταν είναι άρτια έχει αποτέλεσμα 0.