

lab4: if/switch/λογικές εκφράσεις

Προετοιμασία:

1. Πλοηγηθείτε στον κατάλογο **ce120**
2. Κατασκευάστε μέσα στο ce120 ένα νέο κατάλογο με όνομα **lab4**
3. Κατεβάστε το αρχείο **lab4Thu.tar.gz** και τοποθετήστε τα αρχεία που περιέχει μέσα στον κατάλογο **lab4**.
4. Ολοκληρώστε τις ασκήσεις της εβδομάδας.

Θέματα εργασίας:

- Χρήση εντολής if
- Χρήση εντολής switch
- Σύνθετες λογικές εκφράσεις
- Χρήση σταθερών
- Αναγνωσιμότητα κώδικα (σχόλια, στοίχιση, περιγραφικά ονόματα μεταβλητών, κτλ.)

Έλεγχος ορθότητας

Για κάθε άσκηση θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου. Πέρα από τους ελέγχους που θα κάνετε εσείς, συνιστούμε να τρέξετε τα προγράμματά σας και με τις δικές μας εισόδους, να αποθηκεύσετε τις αντίστοιχες εξόδους σας σε αρχεία με χρήση ανακατεύθυνσης και μετά να τις συγκρίνετε με τις δικές μας εξόδους με χρήση της εντολής diff.

Για παράδειγμα, αν σας έχουμε δώσει το αρχείο εισόδου in και την αντίστοιχη έξοδο std, τρέξτε

```
./lab4a < in > myout.txt
```

για να αποθηκεύσετε την έξοδό σας στο αρχείο myout.txt, και μετά συγκρίνετε

```
diff myout.txt std
```

Αν δεν υπάρχουν διαφορές, δε θα εμφανιστεί τίποτα. Αν υπάρχουν, θα εμφανιστούν ανά γραμμή

Άσκηση 1

Προσθέστε τον κώδικά σας στο αρχείο **lab4a.c** που σας δώσαμε.

Γράψτε ένα πρόγραμμα το οποίο υπολογίζει ένα λογαριασμό ηλεκτρικού ρεύματος με βάση το παρακάτω τιμολόγιο:

Μέχρι και τις πρώτες 2000 KWh χρεώνει **0.0681** ευρώ ανά KWh

Από 2001 KWh και πάνω χρεώνει **0.09227** ευρώ ανά KWh

Για παράδειγμα, αν η κατανάλωση είναι 3000 KWh θα χρεώσει $2000 * 0.0681 + 1000 * 0.09227$ ευρώ.

Επιπλέον, κάνει έκπτωση **15%** σε όσους έχουν πληρώσει εγκαίρως τους δύο τελευταίους λογαριασμούς και έκπτωση **7.5%** σε όσους έχουν πληρώσει μόνο έναν από τους δύο τελευταίους λογαριασμούς.

Για την ακρίβεια, το πρόγραμμά σας πρέπει να κάνει τα εξής:

1. Εκτυπώνει το μήνυμα “**Consumption (KWh):** ” με κενό χαρακτήρα μετά το : και διαβάζει την κατανάλωση σε KWh
2. Εκτυπώνει το μήνυμα “**Last bill on time? (y/n)** ” με κενό χαρακτήρα μετά το) και διαβάζει ένα χαρακτήρα ο οποίος θα πρέπει να είναι ‘y’ αν είχε πληρωθεί εγκαίρως ο προηγούμενος λογαριασμός, διαφορετικά είναι ‘n’.
3. Εκτυπώνει το μήνυμα “**Second to last bill on time? (y/n)** ” με κενό χαρακτήρα μετά το) και διαβάζει ένα χαρακτήρα ο οποίος θα πρέπει να είναι ‘y’ αν είχε πληρωθεί εγκαίρως προ-προηγούμενος λογαριασμός, διαφορετικά είναι ‘n’.
4. Αφού διαβάσει όλα τα δεδομένα, **υπολογίζει την αρχική οφειλή** με βάση το παραπάνω τιμολόγιο.
5. Ελέγχει αν πληρώθηκαν εγκαίρως και οι δύο προηγούμενοι μήνες (χρησιμοποιήστε μία if με σύνθετη συνθήκη) κι αν ναι **υπολογίζει τη σχετική έκπτωση**. Αν όχι, ελέγχει αν πληρώθηκε εγκαίρως έστω ο ένας μήνας (και πάλι μία if με σύνθετη συνθήκη) και **υπολογίζει την αντίστοιχη έκπτωση**. Διαφορετικά, δεν υπάρχει έκπτωση.
6. Αφού ολοκληρωθούν τα βήματα 4-5, **υπολογίζει την τελική οφειλή** και εκτυπώνει χαρακτήρα αλλαγής γραμμής, το μήνυμα:

Charge: X.XX

Discount: Y.YY

Total: Z.ZZ

και χαρακτήρα αλλαγής γραμμής. X.XX είναι η αρχική οφειλή, Y.YY το ποσό (όχι το ποσοστό) της έκπτωσης και Z.ZZ η τελική οφειλή. Οι αριθμοί εμφανίζονται με δύο δεκαδικά ψηφία

Αρχεία εισόδου: **in1a, in2a, in3a**

Αντίστοιχα αρχεία εξόδου: **std1a, std2a, std3a**

Άσκηση 2

Προσθέστε τον κώδικά σας στο αρχείο **lab4b.c** που σας δώσαμε.

Περιγραφή προβλήματος

Γράψτε ένα πρόγραμμα το οποίο διαχειρίζεται τη λειτουργία ενός αυτόματου πωλητή αναψυκτικών. Επιτρέπει στο χρήστη να επιλέγει ένα αναψυκτικό, υπολογίζοντας και εμφανίζοντας στην οθόνη το ποσό πληρωμής. Στο τέλος, δέχεται την πληρωμή, εκτυπώνει τα ρέστα αν υπάρχουν, και τερματίζει.

Το πρόγραμμά σας πρέπει να χρησιμοποιεί εντολή switch όπου ενδείκνυται.

Αλληλεπίδραση με το χρήστη

1. Το πρόγραμμα εκτυπώνει στην οθόνη το παρακάτω μενού:

C. Coke

D. Diet Coke

S. Sprite

W. Water

=>

Στην αρχή κάθε γραμμής βρίσκεται ένας χαρακτήρας tab ('\t').

2. Διαβάζει την επιλογή του χρήστη η οποία πρέπει να είναι ένας χαρακτήρας.
3. Αν ο χρήστης δώσει έναν από τους χαρακτήρες 'C', 'D', 'S', 'W', το πρόγραμμα εκτυπώνει το μήνυμα **"Pay X cents:"** με κενό μετά το : όπου X η τιμή του αντίστοιχου αναψυκτικού. Διαφορετικά το πρόγραμμα εκτυπώνει το μήνυμα **"Invalid selection."**, χαρακτήρα αλλαγής γραμμής και τερματίζει με άμεση χρήση της εντολής return.
4. Αν είχε δοθεί έγκυρη επιλογή, το πρόγραμμα διαβάζει από το πληκτρολόγιο το ποσό πληρωμής του χρήστη, υπολογίζει τα ρέστα κι εφόσον είναι μη μηδενικά εκτυπώνει **"Change: Y cents"** όπου Y τα ρέστα σε λεπτά. Ακολουθεί χαρακτήρας αλλαγής γραμμής.

Επιπλέον πληροφορίες

Οι τιμές των προϊόντων που πωλούνται από τη μηχανή είναι:

Τα κουτάκια κοκα-κόλας (Coke, Diet Coke) κοστίζουν 60 λεπτά το καθένα.

Η Sprite κοστίζει 65 λεπτά.

Τα μπουκάλια νερό κοστίζουν 75 λεπτά.

Αρχεία εισόδου: **in1b**, **in2b**, **in3b**

Αντίστοιχα αρχεία εξόδου: **std1b**, **std2b**, **std3b**

Αποστολή Προγραμμάτων για σχολιασμό και βαθμολόγηση

Πλοηγηθείτε στον κατάλογο **lab4**. Θα πρέπει να βρίσκονται μέσα τα αρχεία **lab4a.c** και **lab4b.c**

Κατασκευάστε με χρήση `mkdir` ένα κατάλογο με όνομα `lab4_επώνυμο1_AEM1_επώνυμο2_AEM2` (ΜΕ ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ) όπου επώνυμο1, AEM1 αναφέρονται στο ένα μέλος της ομάδας και επώνυμο2, AEM2 στο δεύτερο μέλος. Για παράδειγμα, αν τα δύο μέλη της ομάδας ήταν οι Ντουφεξή με AEM 1234 και Θάνος με AEM 5678, θα γράφατε:

```
mkdir lab4_Doufexi_1234_Thanos_5678
```

Αντιγράψτε τα αρχεία C μέσα στον κατάλογο `lab4_επώνυμο1_AEM1_επώνυμο2_AEM2` με χρήση της εντολής `cp`. Για παράδειγμα, για την παραπάνω ομάδα θα γράφατε:

```
cp lab4a.c lab4_Doufexi_1234_Thanos_5678/
```

```
cp lab4b.c lab4_Doufexi_1234_Thanos_5678/
```

Πακετάρετε τον κατάλογο. Η εντολή (ακολουθώντας το παραπάνω παράδειγμα) είναι:

```
tar czvf lab4_Doufexi_1234_Thanos_5678.tar.gz lab4_Doufexi_1234_Thanos_5678/
```

Το ένα μέλος της ομάδας:

- μπαίνει στο λογαριασμό του στο eclass,
- πηγαίνει στην ενότητα Εργασίες,
- κάνει click στο όνομα lab4
- Ανεβάζει το .tar.gz αρχείο μέσω του Browse και πατά Αποθήκευση.
- **ΕΛΕΓΧΕΙ ΟΤΙ ΕΣΤΕΙΛΕ ΤΟ ΣΩΣΤΟ .TAR.GZ ΑΡΧΕΙΟ ΚΑΙ ΟΤΙ ΑΥΤΟ ΠΕΡΙΕΧΕΙ ΤΑ ΣΩΣΤΑ .C ΑΡΧΕΙΑ!**

ΑΝ ΤΕΛΕΙΩΣΑΤΕ ΚΑΙ ΥΠΑΡΧΕΙ ΑΚΟΜΗ ΧΡΟΝΟΣ ΜΕΧΡΙ ΤΗ ΛΗΞΗ ΤΟΥ ΤΜΗΜΑΤΟΣ:

Extra Άσκηση

Όταν παίρνουμε δεδομένα από το πληκτρολόγιο, συχνά θέλουμε να επιβεβαιώνουμε ότι έχουν σωστό εύρος πριν τα χρησιμοποιήσουμε.

Προσθέστε κώδικα στην άσκηση 1 έτσι ώστε:

- Όσο ο χρήστης δίνει αρνητικό αριθμό για KWh, το πρόγραμμα εκτυπώνει αλλαγή γραμμής, το μήνυμα **“Positive values only! Try again:”** με κενό μετά το : , και ξαναδιαβάζει το ποσό.
- Όσο ο χρήστης δίνει γράμμα διάφορο των y, n ως απάντηση στα βήματα 2 και 3, το πρόγραμμα ξαναεκτυπώνει τα μηνύματα αυτών των βημάτων και ξαναδιαβάζει τις απαντήσεις.
- Σκεφτείτε τουλάχιστον 2 τρόπους να μεταβάλλετε το πρόγραμμα έτσι ώστε να δέχεται και Y, N ως έγκυρες απαντήσεις.