ΕΡΓΑΣΤΗΡΙΟ 4

Θέμα: Επαναλήψεις, σταδιακή ανάπτυξη κώδικα

Μορφοποίηση κώδικα

Ακολουθήστε τις υποδείξεις των εκφωνήσεων lab1-lab3 και του φυλλαδίου με τίτλο "**Οδηγός γραφής ευανάγνωστου κώδικα**" που βρίσκεται στο eclass, στην ενότητα Έγγραφα->Φυλλάδια/Εγχειρίδια.

Σταδιακή ανάπτυξη κώδικα

Ένα πρόγραμμα με πολλές απαιτήσεις είναι δύσκολο να γραφτεί όλο μαζί, γιατί αν γραφτεί ολόκληρο και δε λειτουργεί είναι δύσκολο να βρεθεί πού είναι το πρόβλημα. Γι' αυτό οι προγραμματιστές ξεκινούν πάντα από ένα μικρό πρόγραμμα που δουλεύει και το εμπλουτίζουν σταδιακά, δοκιμάζοντάς το σε κάθε βήμα.

Η πρώτη άσκηση του lab4 χωρίζεται σε τέσσερα στάδια. Κάθε στάδιο προσθέτει κάτι νέο στο προηγούμενο. Αναπτύξτε τον κώδικα με τον τρόπο που παρουσιάζεται. Στο τέλος κάθε σταδίου και πριν προχωρήσετε στο επόμενο στάδιο, ελέγξτε εάν αυτό που γράψατε λειτουργεί σωστά. Προτείνουμε κάποια στοιχειώδη τεστ, αλλά σκεφτείτε και δικά σας.

Απαιτήσεις:

- Απαγορεύονται **αυστηρά** η χρήση goto και η χρήση πινάκων σε αυτό το εργαστήριο.
- Χρησιμοποιήστε την καταλληλότερη δομή επανάληψης για κάθε περίπτωση.

Οδηγίες για την υποβολή στο autolab

1. Κατασκευάστε ένα zip με όνομα **`lab4.zip`** που να περιέχει ΜΟΝΟ τα αρχεία `lab4a.c` και `lab4b.c`
2. Ακολουθήστε τη διαδικασία σχηματισμού ομάδας για το lab4.
3. Υποβάλετε το αρχείο **`lab4.zip`** στο autolab.

ΠΡΟΣΟΧΗ: Έχετε όριο 10 υποβολών. Χρησιμοποιήστε τα αρχεία ελέγχου ώστε να επιβεβαιώσετε ότι το πρόγραμμα σας λειτουργεί σωστά ΠΡΙΝ υποβάλετε στο autolab.

Άσκηση 1

Αποθηκεύστε το πρόγραμμά σας σε αρχείο με όνομα **`lab4a.c`**

Γράψτε ένα πρόγραμμα που προσομοιώνει συναλλαγές σε ένα τραπεζικό λογαριασμό.

Σας δίνονται τα εξής δεδομένα:

- Δ1: Ο λογαριασμός έχει αρχικό υπόλοιπο 1200.50 ευρώ και το επιτόκιο είναι 2%
- Δ2: Το ποσό που επιτρέπεται να ληφθεί ή κατατεθεί στο λογαριασμό σε μια συναλλαγή ανάληψης ή κατάθεσης πρέπει να είναι στο διάστημα [20, 500] ευρώ.
- Hint: Γράψτε το πρόγραμμα σας ώστε να είναι εύκολο να αλλαχθούν μεταγενέστερα αυτές οι ποσότητες.

Επιπλέον:

- Ως συναλλαγές ορίζονται οι λειτουργίες ανάληψης, κατάθεσης, ερώτησης υπολοίπου και υπολογισμού τόκου. Δεν υπολογίζονται προσπάθειες ανάληψης/κατάθεσης ποσού εκτός του αποδεκτού εύρους.
- Τα αρχικά W, D, B, I, X σημαίνουν Withdrawal (ανάληψη), Deposit (κατάθεση), Balance (υπόλοιπο), Interest (προσθήκη τόκου), Exit (έξοδος).

Στάδιο 1: Ο βασικός βρόχος

Σε επανάληψη, το πρόγραμμα:

1. Εκτυπώνει το μήνυμα "`Action WDBIX\n`" και διαβάζει την επιλογή του χρήστη (char).
2. Αν η επιλογή είναι '`X`', η επανάληψη τερματίζει.
3. Διαφορετικά, με χρήση switch ελέγχει την επιλογή του χρήστη:
   - Αν είναι '`B`', εκτυπώνει το μήνυμα "`Y\n`" όπου `Y` το υπόλοιπο του λογαριασμού με 2 δεκαδικά ψηφία
   - Αν είναι '`I`', υπολογίζει τον τόκο επί του τρέχοντος υπολοίπου και τον προσθέτει στο υπόλοιπο.
   - Σε κάθε άλλη περίπτωση, εκτυπώνει το μήνυμα "`Error (E)\n`" όπου `E` η επιλογή του χρήστη.

Έλεγχος σταδίου 1:

Εκτελέστε το πρόγραμμα σας με επιλογές `B`, `I`, `B`, `K`, `X`.

Θα πρέπει να εκτυπωθεί το αρχικό υπόλοιπο (1200.50), το υπόλοιπο μετά την προσθήκη τόκου (1224.51) και το μήνυμα `Error (K)`

Στάδιο 2: Κατάθεση

Προσθέστε στη switch τη διαχείριση της επιλογής 'D'. Το πρόγραμμα:

1. Εκτυπώνει το μήνυμα "Deposit X-Y\n" όπου X και Y το ελάχιστο και μέγιστο αντίστοιχα ποσό κατάθεσης, χωρίς δεκαδικά ψηφία.
2. Διαβάζει το ποσό κατάθεσης.
3. Όσο αυτό δεν είναι εντός των επιτρεπτών ορίων το βήμα (εκτύπωση + ανάγνωση) επαναλαμβάνεται.
4. Εφόσον δοθεί σωστό ποσό, το υπόλοιπο ανανεώνεται.

Έλεγχος σταδίου 2:

Εκτελέστε το πρόγραμμα σας με εισόδους D, 100, B, D, 10, 500.1, 20, B, X.

Θα πρέπει στο τέλος το υπόλοιπο να είναι 1320.50. Τα ποσά 10 και 500.1 δε γίνονται δεκτά γιατί είναι εκτός του επιτρεπτού εύρους. Δοκιμάστε κι άλλες οριακές τιμές (π.χ. 20, 500).

Στάδιο 3: Ανάληψη

Προσθέστε στη switch τη διαχείριση της επιλογής 'W'. Το πρόγραμμα:

1. Εκτυπώνει το μήνυμα "Withdraw X-Y\n" όπου X και Y το ελάχιστο και μέγιστο αντίστοιχα ποσό ανάληψης, χωρίς δεκαδικά ψηφία.
2. Διαβάζει το ποσό ανάληψης.
3. Όσο αυτό δεν είναι εντός των επιτρεπτών ορίων το βήμα (εκτύπωση + ανάγνωση) επαναλαμβάνεται.
4. Εφόσον δοθεί ποσό εντός των ορίων, ελέγχεται η επάρκεια του υπολοίπου. Αν το υπόλοιπο επαρκεί, ολοκληρώνεται η ανάληψη και ανανεώνεται το υπόλοιπο. Διαφορετικά εκτυπώνεται το μήνυμα "Low balance (X)\n" όπου X το διαθέσιμο υπόλοιπο με 2 δεκαδικά ψηφία και η συναλλαγή θεωρείται αποτυχημένη.

Pro tip: Θα μπείτε στον πειρασμό να κάνετε copy+paste τον κώδικα του σταδίου 2 λόγω των ομοιοτήτων της λειτουργίας. ΠΡΟΣΟΧΗ: Αποφύγετέ το γιατί είναι πολύ εύκολο να ξεχάσετε να αλλάξετε ό,τι χρειάζεται για τη νέα περίπτωση.

Έλεγχος σταδίου 3:

Εκτελέστε το πρόγραμμα σας με εισόδους W, 100, W, 19.9, 501, 500, W, 500, W, 500,B, X.

Θα πρέπει στο τέλος το υπόλοιπο να είναι 100.50. Τα ποσά 19.9 και 501 δε γίνονται δεκτά γιατί είναι εκτός του επιτρεπτού εύρους ενώ η τελευταία ανάληψη 500 ευρώ έχει ως αποτέλεσμα την εκτύπωση του μηνύματος Low balance (100.50).

Στάδιο 4: Τελική αναφορά

Προσθέστε στο πρόγραμμα σας κατάλληλους μετρητές ώστε να μετρά το σύνολο των συναλλαγών και το σύνολο των επιτυχημένων συναλλαγών. Μετά την έξοδο από την κύρια επανάληψη, το πρόγραμμα πρέπει να εκτυπώνει το παρακάτω μήνυμα (με χαρακτήρα '\n' στο τέλος κάθε γραμμής):

```
Balance: B euros
Transactions: T
Failed: F%
```

όπου B το τελικό υπόλοιπο με 2 δεκαδικά ψηφία, T το σύνολο των συναλλαγών και F το ποσοστό των αποτυχημένων συναλλαγών με 1 δεκαδικό ψηφίο. Αν δεν έχουν γίνει συναλλαγές, η τιμή του F είναι μηδέν (γιατί χρειάζεται να ελεγχθεί ρητά αυτό;)

Έλεγχος σταδίου 4:

Για την είσοδο που δώσατε κατά τον έλεγχο του σταδίου 3, το πλήθος συναλλαγών είναι 5 και το ποσοστό αποτυχημένων συναλλαγών είναι 20.0%

Άσκηση 2

Αποθηκεύστε το πρόγραμμά σας σε αρχείο με όνομα **`lab4b.c`**

Γράψτε ένα πρόγραμμα το οποίο εκτυπώνει ένα ορθογώνιο με εναλλάξ χαρακτήρες.

Το πρόγραμμα λειτουργεί ως εξής:

1. Εκτυπώνει "`Enter rows and columns [1-20]:\n`" και διαβάζει δύο ακέραιους: πλήθος γραμμών και πλήθος στηλών. Όσο οποιοδήποτε πλήθος δεν ανήκει στο διάστημα [1, 20], το βήμα 1 επαναλαμβάνεται.
2. Εκτυπώνει "`Enter 2 distinct characters:\n`" και διαβάζει 2 χαρακτήρες χωρίς κενό ανάμεσα τους. Όσο αυτοί είναι ίδιοι, εκτυπώνει "`Try again!\n`" και το βήμα 2 επαναλαμβάνεται.
3. Με χρήση αποκλειστικά της συνάρτησης putchar (όχι printf σε αυτό το βήμα!) εμφανίζει στην οθόνη ένα σχήμα που σχηματίζεται από τους δύο χαρακτήρες ως εξής:
   - Κάθε γραμμή έχει τόσες εμφανίσεις του κάθε χαρακτήρα όσο το πλήθος στηλών, τοποθετώντας τους εναλλάξ, χωρίς κενό ανάμεσα, και χαρακτήρα αλλαγής γραμμής στο τέλος.
   - Η πρώτη γραμμή ξεκινά από τον πρώτο χαρακτήρα που δόθηκε στο βήμα 2 και κάθε επόμενη γραμμή ξεκινά από τον χαρακτήρα που είναι διαφορετικός από αυτόν που ξεκίνησε η προηγούμενη.

Υπόδειξη: Χρησιμοποιήστε μια μεταβλητή για να αποθηκεύσετε τον χαρακτήρα προς εκτύπωση και σκεφτείτε πώς μπορείτε να υλοποιήσετε την εναλλαγή.

Παράδειγμα:

Για εισόδους `4 5` στο βήμα 1 και `-X` στο βήμα 2, θα εμφανιστεί στην οθόνη:

```
-X-X-X-X-X
X-X-X-X-X-
-X-X-X-X-X
X-X-X-X-X-
```