

## lab4: Εντολές Ελέγχου

### Προετοιμασία:

1. Πλοηγηθείτε στον κατάλογο lab4 ο οποίος πρέπει να βρίσκεται μέσα στον κατάλογο ce120.
2. Ολοκληρώστε τις ασκήσεις της εβδομάδας. Τα αρχεία που θα γράψετε πρέπει να αποθηκευτούν στον κατάλογο lab4 που δημιουργήσατε προηγουμένα.
  - Θυμίζουμε πως η εντολή για να κάνετε compile ένα αρχείο με όνομα **lab4a.c** είναι:

```
gcc -Wall -g lab4a.c -o lab4a
```

Αν δεν υπάρχουν λάθη, θα δημιουργηθεί το εκτελέσιμο αρχείο **lab4a** το οποίο μπορείτε να εκτελέσετε γράφοντας **./lab4a**

### Θέματα εργασίας:

- Χρήση εντολών ελέγχου
- Χρήση τελεστών
- Χρήση σταθερών (πρέπει να σκεφτείτε μόνοι σας ποιες ποσότητες πρέπει να οριστούν ως σταθερές)
- Χρήση μαθηματικών συναρτήσεων
- Αναγνωσιμότητα κώδικα (σχόλια, στοίχιση, περιγραφικά ονόματα μεταβλητών, κτλ.)

### Ελεγχος 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 in1.txt και την αντίστοιχη έξοδο std1.txt, τρέξτε

```
./lab4a < in1.txt > out1.txt
```

για να αποθηκεύσετε την έξοδό σας στο αρχείο out1.txt, και μετά συγκρίνετε

```
diff out1.txt std1.txt
```

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

## Άσκηση 1

Το πρόγραμμα που θα γράψετε για την άσκηση 1 πρέπει να το αποθηκεύσετε σε αρχείο με όνομα **lab4a.c** μέσα στον κατάλογο **lab4**.

Σε ένα parking αυτοκινήτων το οποίο δέχεται αυτοκίνητα η τιμολογιακή πολιτική διαμορφώνεται ως εξής:

- Πρώτες δύο ώρες: 2.5 ευρώ η ώρα
- Εφόσον παραμείνει για περισσότερες από δύο ώρες και λιγότερες από 24 ώρες, το συνολικό κόστος διαμορφώνεται σε 1.4 ευρώ η ώρα.
- Εφόσον παραμείνει για 24 ώρες και πάνω, το συνολικό κόστος διαμορφώνεται σε 0.75 ευρώ η ώρα.

Σε όλες τις παραπάνω τιμές οι πελάτες έχουν έκπτωση 15% εφόσον έχουν επισκεφθεί το parking τουλάχιστον μία φορά στο παρελθόν.

Γράψτε ένα πρόγραμμα σε γλώσσα C που κάνει τα εξής:

1. Εκτυπώνει το μήνυμα: **"Time: "** (υπάρχει ένα κενό μετά το χαρακτήρα ':')
2. Ο χρήστης εισάγει έναν ακέραιο που αντιπροσωπεύει τις ώρες που παρέμεινε στο parking το όχημα του.
3. Εκτυπώνει το μήνυμα: **"Old Customer: "** (υπάρχει ένα κενό μετά το χαρακτήρα ':')
  - ◆ Ο χρήστης εισάγει ένα χαρακτήρα όπου εφόσον είναι 'Y' ή 'y' πρόκειται για παλιό πελάτη, διαφορετικά πρόκειται για νέο πελάτη.
4. Υπολογίζει το συνολικό κόστος με βάση την παραπάνω πολιτική.
5. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και αμέσως μετά το ακόλουθο μήνυμα:  
**Time: HH hours**  
**Discount: XXX.XX**  
**Cost: YYY.YY**  
όπου **HH** οι ώρες που παρέμεινε στο parking το όχημα με πλάτος 2 ψηφίων, **XXX.XX** η έκπτωση και **YYY.YY** το συνολικό κόστος εκτυπωμένα με ακρίβεια δύο δεκαδικών ψηφίων.
6. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής**.

Αρχεία εισόδου: **a\_in1.txt, a\_in2.txt, a\_in3.txt**

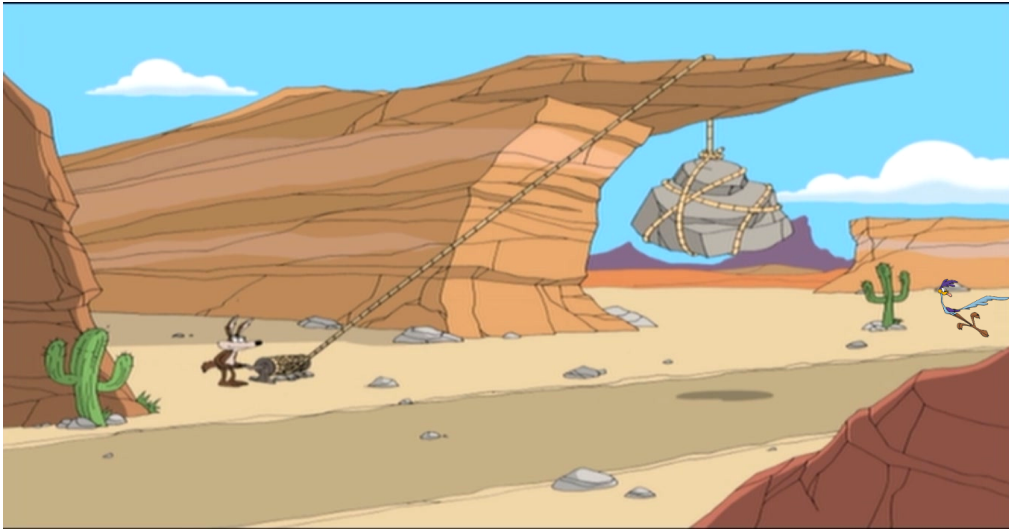
Αντίστοιχα αρχεία εξόδου: **a\_std1.txt, a\_std2.txt, a\_std3.txt**

## Άσκηση 2

Το πρόγραμμα που θα γράψετε για την άσκηση 2 πρέπει να το αποθηκεύσετε σε αρχείο με όνομα **lab4b.c** μέσα στον κατάλογο lab4.

Διαβάστε προσεκτικά ΟΛΗ την εκφώνηση πριν ξεκινήσετε.

Το κογιότ έχει κουραστεί να κυνηγάει το μπιπ-μπιπ. Όμως, έχει μια νέα μέθοδο για να το πιάσει επιτέλους! Θα ανεβάσει ένα βράχο ψηλά με μία τροχαλία και όταν δει το μπιπ-μπιπ να έρχεται, θα αφήσει το βράχο να πέσει...



Καλείστε να γράψετε ένα πρόγραμμα που θα προβλέψει αν τελικά το κογιότ θα φάει σνιτσελ μπιπ-μπιπ ή όχι. Το πρόγραμμα:

1. Εμφανίζει στην οθόνη το μήνυμα "**Cliff height:** " (υπάρχει κενό μετά τον χαρακτήρα ':').
2. Διαβάζει έναν αριθμό κινητής υποδιαστολής ο οποίος είναι το ύψος του βράχου σε μέτρα.
3. Εμφανίζει στην οθόνη το μήνυμα "**Roadrunner distance:** " (υπάρχει κενό μετά τον χαρακτήρα ':').
4. Διαβάζει έναν αριθμό κινητής υποδιαστολής ο οποίος εκφράζει πόσα μέτρα μακριά είναι το μπιπ-μπιπ από το σημείο που θα προσγειωθεί ο βράχος, τη στιγμή που το κογιότ αφήνει το βράχο.
5. Υπολογίζει σε πόση ώρα το μπιπ-μπιπ θα βρίσκεται στο σημείο που θα πέσει ο βράχος.
6. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και την **ώρα** που υπολόγισε στο βήμα 5 με 4 δεκαδικά ψηφία.
7. Υπολογίζει πόση ώρα θα κάνει ο βράχος να πέσει.
8. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και την **ώρα** που υπολόγισε στο βήμα 7 με 4 δεκαδικά ψηφία.
9. Αν οι χρόνοι δε διαφέρουν περισσότερο από ένα δέκατο του δευτερολέπτου (+/-), το πρόγραμμα εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και "**SPLAT!**" (το μπιπ-μπιπ έγινε σνίτσελ), διαφορετικά εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και "**Zoom!**".
10. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής**.

**Διαβάστε τις διευκρινίσεις που ακολουθούν πριν αρχίσετε να γράφετε κώδικα!**

**Διευκρινίσεις:**

**[Βήμα 5]** Η σχέση ανάμεσα στην απόσταση και στο χρόνο για το μπιτ-μπιτ που κινείται με σταθερή ταχύτητα δίνεται από τον τύπο  $S = v * t$  όπου S:απόσταση, v: ταχύτητα, t: χρόνος. Η ταχύτητα του μπιτ-μπιτ είναι ίση με **32.18 km/h**. Θα πρέπει να γράψετε κώδικα που υπολογίζει το χρόνο με βάση την απόσταση και την ταχύτητα αφού πρώτα μετατρέψει την ταχύτητα από km/h σε m/sec.

**[Βήμα 7]** Η σχέση ανάμεσα στην απόσταση και το χρόνο για το βράχο δίνεται από τον τύπο  $S = \frac{1}{2} * g * t^2$  όπου S η απόσταση, t ο χρόνος και g η σταθερά επιτάχυνσης της βαρύτητας, η οποία έχει την τιμή **9.8m/sec<sup>2</sup>**. Θα πρέπει να γράψετε κώδικα που υπολογίζει το χρόνο με δεδομένες τις υπόλοιπες ποσότητες του τύπου.

Η μαθηματική συνάρτηση υπολογισμού τετραγωνικής ρίζας λέγεται **sqrt**, για παράδειγμα το **sqrt(x)** υπολογίζει την τετραγωνική ρίζα της τιμής που είναι αποθηκευμένη στη μεταβλητή x η οποία είναι τύπου double.

**[Βήμα 9]** Με άλλα λόγια, για να πέσει ο βράχος στο μπιτ-μπιτ, η διαφορά των χρόνων πρέπει να είναι μεταξύ -0.1 και 0.1 ή, ισοδύναμα, η απόλυτη τιμή της διαφοράς να είναι μικρότερη του 0.1

**Προσοχή:** ΜΗ χρησιμοποιήσετε ονόματα μεταβλητών του ενός γράμματος (g, t, κτλ.) αλλά περιγραφικά ονόματα.

Αρχεία εισόδου: **b\_in1.txt, b\_in2.txt, b\_in3.txt**

Αντίστοιχα αρχεία εξόδου: **b\_std1.txt, b\_std2.txt, b\_std3.txt**

**Προαιρετική άσκηση για όσους τελειώσουν νωρίς:**

Γράψτε ένα πρόγραμμα το οποίο ζητά δύο ημερομηνίες και υπολογίζει πόσες μέρες βρίσκονται ανάμεσά τους.

Το πρόγραμμα τυπώνει αρχικά το μήνυμα **"Enter period (D/M/Y - D/M/Y): "**

Τα δεδομένα θα πρέπει να δίνονται από το χρήστη με τη μορφή : D/M/Y - D/M/Y . Η κάθε τριάδα D/M/Y αναπαριστά ημέρα/μήνα/έτος. Δώστε προσοχή στις εμφανίσεις των χαρακτήρων '/' και '-'.

Το πρόγραμμά σας πρέπει να ελέγχει αν η διαφορά ετών είναι το πολύ δύο\*, καθώς και αν η δεύτερη ημερομηνία είναι μεταγενέστερη της πρώτης. Αν δεν ισχύει κάτι από αυτά, εκτυπώνει σχετικό μήνυμα λάθους και τερματίζει.

Κατόπιν, το πρόγραμμα πρέπει να μετρά όλες τις μέρες ανάμεσα στις δύο ημερομηνίες και να εκτυπώνει το πλήθος τους. Τα δίσεκτα έτη έχουν 366 ημέρες, ενώ τα κανονικά έχουν 365 ημέρες. Ένα έτος είναι δίσεκτο αν διαιρείται ακριβώς με το 4, αλλά όχι με το 100. Κατ' εξαίρεση του παραπάνω κανόνα, αν ένα έτος διαιρείται ακριβώς με το 400, τότε είναι δίσεκτο. Οι μήνες με 30 ημέρες είναι Απρίλιος, Ιούνιος, Σεπτέμβριος, Νοέμβριος.

\* Για μεγαλύτερες διαφορές χρειάζεται χρήση επανάληψης.

**Αποστολή Προγραμμάτων για σχολιασμό και βαθμολόγηση**

- Πλοηγηθείτε στον κατάλογο **lab4**. Θα πρέπει να βρίσκονται μέσα τα αρχεία **lab4a.c** και **lab4b.c**
- Κατασκευάστε να 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 01234 και Θάνος με AEM 05678, θα γράφατε:

**mkdir lab4\_Doufexi\_01234\_Thanos\_05678**

- Αντιγράψτε τα αρχεία C μέσα στον κατάλογο lab4\_επώνυμο1\_AEM1\_επώνυμο2\_AEM2
- Πακετάρετε τον κατάλογο (compress here as tar.gz).
- Το ένα μέλος της ομάδας:
  - μπαίνει στο λογαριασμό του στο eclass,
  - πηγαίνει στην ενότητα Εργασίες,
  - κάνει click στην εργασία lab4 για το συγκεκριμένο τμήμα
  - **ΠΡΟΣΟΧΗ!!!!** Στο πεδίο Ομάδα Χρηστών επιλέγει τη σωστή ομάδα (με βάση τα ονόματα των μελών)
  - Ανεβάζει το .tar.gz αρχείο μέσω του Browse **και πατά Αποθήκευση.**
  - **ΠΡΟΣΟΧΗ!!!!** Επιβεβαιώνει ότι ανέβασε το σωστό αρχείο.