

## **ΕΡΓΑΣΤΗΡΙΟ 3**

### **ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012**

#### **Στόχοι**

- Δομές ελέγχου και επανάληψης
- Μεταβλητές/σταθερές
- Μορφοποιημένη έξοδος
- Ανάπτυξη αλγορίθμων

#### **Πριν ξεκινήσετε**

##### **Βήμα 1:**

Πηγαίνετε στο φάκελο ce120 και κατασκευάστε μέσα σε αυτόν ένα φάκελο με όνομα lab3

##### **Βήμα 2:**

Ανοίξτε το Kate και κατασκευάστε δύο νέα αρχεία. Το ένα πρέπει να έχει όνομα lab3a.c και το άλλο lab3b.c. Πρέπει να είναι αποθηκευμένα στο φάκελο lab3 που κατασκευάσατε στο προηγούμενο βήμα.

##### **Βήμα 3:**

Κάντε τις παρακάτω ασκήσεις. Σας θυμίζουμε πως η εντολή για να κάνετε compile ένα αρχείο με όνομα lab3.c είναι : **gcc -g -Wall lab3.c -o lab3**

Θα σας φανούν χρήσιμες οι νέες διαφάνειες (5-9) για printf/scanf που προστέθηκαν στις σημειώσεις του δεύτερου φροντιστηρίου. Κάντε click [εδώ](#)

## Άσκηση 1: Δομές ελέγχου κι επανάληψης, μεταβλητές, σταθερές, μορφοποιημένη έξοδος

Το πρόγραμμα που θα γράψετε για την άσκηση 1 πρέπει να το αποθηκεύσετε στο αρχείο **lab3a.c**

Γράψτε ένα πρόγραμμα το οποίο βρίσκει αν το κογιότ ή κάποιο άλλο κυνηγητικό ζώο θα καταφέρει να πιάσει το γνωστό μπιπ-μπιπ (roadrunner) μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Το πρόγραμμα λειτουργεί ως εξής:



1. Εκτυπώνει στην οθόνη το μήνυμα "Hunter : "
2. Διαβάζει από το πληκτρολόγιο τον τύπο του κυνηγού: T αν πρόκειται το κογιότ (coyote), R αν πρόκειται για κούγκαρ (cougar), ή H αν πρόκειται για τσιταχ (cheetah). Αν δοθεί οποιοσδήποτε άλλος τύπος, το πρόγραμμα επαναλαμβάνει τα βήματα 1 και 2.
3. Εκτυπώνει στην οθόνη το μήνυμα "Initial distance: " και διαβάζει την αρχική απόσταση ανάμεσα στον κυνηγό και το μπιπ-μπιπ. Η απόσταση δίνεται σε μέτρα. Εννοείται πως το μπιπ-μπιπ βρίσκεται πιο μπροστά από τον κυνηγό και όταν αρχίσει το κυνήγι, και τα δύο ζώα θα πηγαίνουν προς την ίδια κατεύθυνση. Δε χρειάζεται να ελέγξετε αν οι τιμή είναι έγκυρη.
4. Εκτυπώνει στην οθόνη το μήνυμα "Enter time: " και διαβάζει το χρόνο που θα διαρκέσει το κυνήγι. Ο χρόνος δίνεται στη μορφή HH:MM όπου HH η ώρα στην κλίμακα 0-23 και MM τα λεπτά στην κλίμακα 0-59. Και τα δύο νούμερα είναι ακέραιοι.
5. Για να βρείτε αν τελικά ο κυνηγός θα πιάσει το μπιπ-μπιπ, πρέπει να υπολογίσετε την απόσταση που θα μπορεί να έχει διανύσει ο καθένας τους μέσα σε αυτό το χρονικό διάστημα. Οι ταχύτητες των προαναφερθέντων ζώων είναι :
  - Coyote: 69.2 km/h
  - Cougar: 48.28 km/h
  - Cheetah: 112.65 km/h
  - Roadrunner: 32.18 km/h
6. Το πρόγραμμα στο τέλος εκτυπώνει τα παρακάτω αποτελέσματα:

Hunter	WWWWW
Distance diff	XX.XX
Result	Y

Υπάρχει μια κενή γραμμή πάνω από το Hunter.

WWWWW είναι ο τύπος του κυνηγού (T ή R ή H) και πιάνει 5 θέσεις

XX.XX είναι η διαφορά στην απόσταση ανάμεσα στον οδηγό και το θήραμα, στο τέλος του χρόνου. Εκφράζεται σε χιλιόμετρα, με δύο ακέραια και δύο δεκαδικά ψηφία

Y είναι μια φράση που εκφράζει το αποτέλεσμα όσον αφορά τον κυνηγό. Αν έπιασε το μπιπ-μπιπ, τότε η φράση είναι "Munch munch" ενώ αν δεν το έπιασε, η φράση είναι "Pant pant".

Παράδειγμα εκτέλεσης (αυτά που γράφει ο χρήστης φαίνονται με μπλε χρώμα):

```
Hunter: P
Hunter: T
Initial distance: 500
Enter time: 0:10
```

```
Hunter          T
Distance diff    6.67
Result          Munch munch
```

**Το πρόγραμμά σας πρέπει να χρησιμοποιεί κατάλληλες μεταβλητές, και σταθερές όπου χρειάζεται.**

Πρέπει να κάνει compile χωρίς errors ή warnings και να τρέχει σωστά για διάφορες τιμές εισόδου.

## **Άσκηση 2: Τελεστές, εκφράσεις, αλγόριθμοι**

Το πρόγραμμα που θα γράψετε για την άσκηση 2 πρέπει να το αποθηκεύσετε στο αρχείο **lab3b.c**

Γράψτε ένα πρόγραμμα το οποίο

- Εκτυπώνει στην οθόνη το μήνυμα "Enter number:"
- Διαβάζει έναν ακέραιο αριθμό. Δε γνωρίζουμε πόσα ψηφία έχει
- Αποσπά ένα-ένα τα ψηφία του, με αντίστροφη σειρά (από το τελευταίο προς το πρώτο) και κατασκευάζει από αυτά ένα νέο αριθμό (ο οποίος τελικά θα έχει τα αντίστροφα ψηφία από τον αρχικό).
- Εκτυπώνει το νέο αριθμό

Για παράδειγμα, αν το πρόγραμμα διαβάσει το 256 τότε θα πρέπει να κατασκευάζει τον αριθμό 652.

Μπορείτε να υποθέσετε ότι ο αριθμός που θα δοθεί είναι θετικός.

## Αποστολή του προγράμματος για διόρθωση και σχολιασμό

Η αποστολή γίνεται από τον οδηγό.

Για να στείλετε το πρόγραμμά σας για βαθμολόγηση, ανοίξτε ένα τερματικό (ή χρησιμοποιήστε αυτό του Kate), πηγαίνετε στο φάκελο lab3, και χρησιμοποιήστε την εντολή `mkdir` για να κατασκευάσετε ένα νέο φάκελο με βάση τα ονόματα των δύο μελών της ομάδας. Για την ακρίβεια, το όνομα του φακέλου πρέπει να είναι (με λατινικούς χαρακτήρες) :

`επώνυμο1_AM1_επώνυμο2_AM2`

Για παράδειγμα, `mkdir doufexi_1234_antonopoulos_5678`

AM είναι ο τετραψήφιος αριθμός μητρώου σας.

**Αντιγράψτε** το lab3a.c μέσα στο νέο φάκελο

`(cp lab3a.c doufexi_1234_antonopoulos_5678)`

**Αντιγράψτε** το lab3b.c μέσα στο νέο φάκελο

`(cp lab3b.c doufexi_1234_antonopoulos_5678)`

Τέλος **πακετάρετε** και συμπίεστε το φάκελο με το πρόγραμμά σας. Η εντολή είναι

`tar czf επώνυμο1_AM1_επώνυμο2_AM2.tgz επώνυμο1_AM1_επώνυμο2_AM2`

Θα πρέπει να δημιουργηθεί το αρχείο με όνομα `επώνυμο1_AM1_επώνυμο2_AM2.tgz`

Αν εμφανιστεί κάποιο μήνυμα λάθους, ζητήστε βοήθεια!

**Στείλτε** ένα email

- στη διεύθυνση: **ce120lab@gmail.com**
- με τίτλο (subject): **CE120 lab3 tmimaX**
- **CC** στον πλοηγό, και
- συνημμένο (attached) το αρχείο: `επώνυμο1_AM1_επώνυμο2_AM2.tgz`

**X** είναι το τμήμα σας (1, 2, 3, 4 ή 5)

Αφού στείλετε το email, **επιβεβαιώστε** ότι είχε σωστό attachment κι ότι έλαβε αντίγραφο και ο πλοηγός.

Ο πλοηγός μπορεί να ξεπακετάρει το `tgz` αρχείο στο δικό του υπολογιστή με την εντολή

`tar xzf επώνυμο1_AM1_επώνυμο2_AM2.tgz`

**ΜΗΝ ΞΕΧΑΣΕΤΕ ΝΑ ΚΑΝΕΤΕ ΚΑΙ ΟΙ ΔΥΟ LOGOUT ΠΡΙΝ ΦΥΓΕΤΕ!!**