

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΔΠΜΣ – ΡΟΗ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ  
Η γλώσσα προγραμματισμού C

ΕΡΓΑΣΤΗΡΙΟ 1: Εισαγωγή στην C – Βασικά στοιχεία

Μάρτιος 2019

Ενδεικτική ημερομηνία παράδοσης: 27 Μαρτίου 2019

Το σημερινό εργαστήριο αποτελεί μια εισαγωγή στο περιβάλλον *Code::Blocks* που χρησιμοποιούμε<sup>1</sup>, και έχει ως βασικά θέματα την πρώτη επαφή με ένα πρόγραμμα C και τη γνωριμία με τη βασική διαδικασία επικοινωνίας υπολογιστή-χρήστη της γλώσσας C, τις απλές βαθμωτές μεταβλητές και τις βασικές δομές ελέγχου της γλώσσας C. Ως βαθμωτές (scalar) εννοούμε τις μεταβλητές που δεν έχουν διάσταση, δηλαδή αυτές που δεν ορίζονται ως πίνακες. Θα μελετήσουμε βαθμωτές μεταβλητές απλών τύπων, όπως ακέραιου τύπου και τύπου χαρακτήρα. Για τα προγράμματα που θα γράψετε, θα χρησιμοποιηθεί ως βασική δομή ελέγχου η εντολή εκτέλεσης υπό συνθήκη *if-else* με κάποια απλή έκφραση συνθήκης.

Τα προγράμματα του εργαστηρίου περιγράφονται αναλυτικά, ώστε να αποκτήσετε εύκολα την πρώτη επαφή με την C και τον τρόπο μετάφρασης και εκτέλεσης των προγραμμάτων που γράφετε. Σιγά-σιγά στα επόμενα εργαστήρια θα μειώνεται η πληροφορία που σας δίνεται και θα πρέπει όλο και περισσότερο να γράφετε μόνοι σας τον κώδικα. Από τις παρακάτω ασκήσεις θα παραδώσετε τις Ασκήσεις 2 και 3 με τον τρόπο που περιγράφεται στο τέλος της εκφώνησης. Προσέξτε ότι η διόρθωση δεν θα γίνει (προφανώς) από εσάς, κι επομένως τα προγράμματά σας θα πρέπει να είναι στοιχισμένα και επαρκώς σχολιασμένα, ώστε να είναι κατανοητά!

*Άσκηση 1: «Κατανόηση και επέκταση του προγράμματος "Hello world!"»*

Εκκινήστε το πρόγραμμα *Code::Blocks*, κάνοντας κλικ στο σχετικό εικονίδιο, ή επιλέγοντάς το από τη λίστα προγραμμάτων του υπολογιστή. Στη συνέχεια ακολουθήστε τις οδηγίες του εγχειριδίου, ώστε να βρείτε το εισαγωγικό πρόγραμμα γλώσσας C που εμφανίζεται μόλις ξεκινάτε μια καινούργια εργασία στο *Code::Blocks*. Όταν σας ζητηθεί, ονομάστε την εργασία *lab0* και ορίστε ως θέση της εργασίας την *Επιφάνεια Εργασίας*, κάποια άλλη θέση στο λογαριασμό σας (αν δουλεύετε στο Πανεπιστήμιο), ή κάποια φορητή συσκευή αποθήκευσης.

Στην αρχή του προγράμματος στο αρχείο *main.c* που δημιουργήθηκε αυτόματα από το *Code::Blocks* δηλώνεται η χρήση της βιβλιοθήκης εισόδου/εξόδου *stdio* με τη γραμμή

```
#include <stdio.h>
```

---

<sup>1</sup> Σε περίπτωση που χρησιμοποιείτε άλλη πλατφόρμα προγραμματισμού C, οι ασκήσεις προσαρμόζονται κατάλληλα για εκείνη την πλατφόρμα.

Μέσω του προεπεξεργαστή της C, η γραμμή αυτή δίνει οδηγία στο μεταγλωττιστή να συμπεριλάβει στη μετάφραση το αρχείο επικεφαλίδων της βιβλιοθήκης *stdio*, το οποίο έχει το όνομα της βιβλιοθήκης με κατάληξη `‘.h’`. Ο προεπεξεργαστής της C αποτελεί κατά κάποιον τρόπο επέκταση της γλώσσας, μια που οι οδηγίες του δεν είναι εντολές της C, αλλά τις χρησιμοποιούμε σα να είναι τέτοιες! Οι οδηγίες του προεπεξεργαστή δίνονται με τη βοήθεια του ειδικού συμβόλου `‘#’`. Η χρήση βιβλιοθηκών δηλώνεται με την οδηγία *include*. Η βιβλιοθήκη *stdio* είναι βιβλιοθήκη συστήματος, γι’ αυτό και το όνομα του αρχείου επικεφαλίδων της περικλείεται σε τριγωνικές αγκύλες. Το ίδιο και η βιβλιοθήκη *stdlib* που δηλώνεται στην επόμενη γραμμή, η οποία όμως δεν θα μας απασχολήσει ακόμα. Μετά τις δηλώσεις βιβλιοθηκών, που συνήθως γράφονται στην αρχή του αρχείου του προγράμματος, ακολουθεί μόνο η δήλωση και ο ορισμός της κύριας συνάρτησης *main()*, κάτι που δε συμβαίνει γενικά, μια που τις πιο πολλές φορές ακολουθούν δηλώσεις μεταβλητών και άλλων συναρτήσεων πριν την *main()*.

Η συνάρτηση *main()* του προγράμματός μας το μόνο που κάνει είναι η εκτύπωση κάποιου μηνύματος και η επιστροφή στο σύστημα που σηματοδοτεί τον τερματισμό του προγράμματος. Μέσα στη βιβλιοθήκη *stdio* υπάρχει η συνάρτηση *printf()*, μια από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις βιβλιοθήκης της C. Θα πρέπει να προσέξτε ότι η εκτύπωση του μηνύματος *"Hello world!\n"* έγινε με κλήση αυτής της συνάρτησης. Η συνάρτηση *printf()* είναι συνάρτηση εξόδου της C και εκτυπώνει συμβολοσειρές, συνήθως στην οθόνη του υπολογιστή. Η εκτύπωση γίνεται με διαμόρφωση που καθορίζεται μέσα από την πρώτη παράμετρο της συνάρτησης. Εδώ δεν έχουμε άλλες παραμέτρους, και η διαμόρφωση είναι προφανής. Η μοναδική παράμετρος – δηλαδή η έκφραση μέσα στις παρενθέσεις – είναι ταυτόχρονα η συμβολοσειρά και η διαμόρφωσή της. Η συνάρτηση *printf()* καλείται μέσα από μια εκφυλισμένη εντολή έκφρασης που δεν περιέχει τίποτε άλλο πέρα από την κλήση. Η επιστροφή στο σύστημα γίνεται με την τελευταία γραμμή του σώματος της συνάρτησης *main()*, και ειδικότερα με την εντολή *return* της C, που όμως δε θα μας απασχολήσει ακόμα. Και οι δύο εντολές της *main()* τερματίζονται με το σύμβολο `‘;’`. Προσέξτε ότι το σώμα της συνάρτησης *main()*, όπως και κάθε άλλης συνάρτησης της C, είναι τοποθετημένο μέσα σε αγκύλες.

Αφού μελετήσετε και κατανοήσετε το πρόγραμμα, συνεχίστε με τη μετάφραση και εκτέλεσή του, με τον τρόπο που περιγράφεται στο εγχειρίδιο. Αν δεν υπάρχει κάποιο σφάλμα, το παράθυρο κονσόλας εκτέλεσης θα ανοίξει και θα εκτυπώσει το ζητούμενο μήνυμα!

Στη συνέχεια, αλλάξτε το μήνυμα εκτύπωσης σε ο,τιδήποτε άλλο θέλετε να εμφανίζεται στην οθόνη. Προσέξτε να διατηρήσετε το μήνυμα εντός των διπλών αποστρόφων, και εντός των παρενθέσεων. Επίσης προσέξτε τη χρήση ειδικών χαρακτήρων που εκφράζονται με τη βοήθεια του συμβόλου `‘\’`, καθώς και των χαρακτήρων διαμόρφωσης που εκφράζονται με τη βοήθεια του συμβόλου `‘%’`!

Με την τροποποίηση που κάνατε, το *Code::Blocks* παρατήρησε ότι το αρχείο *main.c* έχει αλλάξει, σημειώνοντας το σύμβολο `‘*’` στο όνομα του αρχείου πάνω από το παράθυρο επεξεργασίας κειμένου, καθώς και το σύμβολο `‘!’` στο εικονίδιο δίπλα στο όνομα του αρχείου στο παράθυρο διαχείρισης εργασιών. Αν προσπαθήσετε να βγάτε από το *Code::Blocks*, αυτό θα σας ειδοποιήσει να αποθηκεύσετε το τροποποιημένο αρχείο, αν φυσικά το θέλετε. Γενικά, όταν αλλάζετε κάποιο αρχείο, καλό είναι να συνηθίσετε να αποθηκεύετε το αρχείο συχνά, μια που συμβαίνουν ατυχήματα και είναι πολύ ενοχλητικό – και ιδιαίτερα κουραστικό – να χάνονται αλλαγές και να πρέπει να ξαναγίνουν από την αρχή! Ευτυχώς το *Code::Blocks* διατηρεί αντίγραφα ασφαλείας

στα αρχεία, ώστε τις περισσότερες φορές να ξαναβρίσκετε αλλαγές που για κάποιο λόγο χάθηκαν.

Ξαναμεταφράστε και εκτελέστε το πρόγραμμα, ώστε να εκτυπωθεί το νέο μήνυμα στο παράθυρο κονσόλας. Κλείστε το παράθυρο κονσόλας. Αποθηκεύστε το αρχείο του προγράμματος και προχωρήστε παρακάτω.

Συνεχίζοντας το προηγούμενο πρόγραμμα, εισάγετε τη δήλωση μιας ακέραιας μεταβλητής τύπου `int`, έστω `age`, ως καθολική, αμέσως κάτω από τις δηλώσεις βιβλιοθηκών, και πριν από τον ορισμό της συνάρτησης `main()`. Η μεταβλητή αυτή έχει ορατότητα σε όλο το πρόγραμμα, δηλαδή μπορεί να γίνει αναφορά σε αυτήν από κάθε εντολή του προγράμματος. Αυτό δε θα συνέβαινε αν είχαμε κάποια δήλωση τοπικής μεταβλητής με το ίδιο όνομα στη `main()` ή σε άλλη συνάρτηση του προγράμματος. Γράψτε τη δήλωση με την ίδια στοίχιση που έχει η προηγούμενη και η επόμενη γραμμή. Έτσι το πρόγραμμα είναι πιο ευανάγνωστο.

Προσέξτε ότι η μεταβλητή που δηλώσατε έχει όνομα που σχετίζεται με αυτό που θέλουμε να περιέχει, το οποίο είναι κάποια ηλικία. Σε περίπτωση που δεν είναι κατανοητό το τι περιέχει μια μεταβλητή από το όνομά της, είναι σκόπιμο να βάζετε κάποιο σχόλιο που να το εξηγεί. Διαφορετικά το πρόγραμμά σας δεν θα διαβάζεται εύκολα, όχι μόνο από κάποιον άλλο, αλλά και από εσάς τους ίδιους. Η επεξήγηση μέσω κατάλληλων ονομάτων και σχολίων είναι πολύ σημαντική προγραμματιστική τακτική, και αυτό θα το καταλάβετε την πρώτη φορά που θα γράψετε κάποιο μεγάλο πρόγραμμα, και ειδικά μόλις διαβάζοντάς το κάποια επόμενη φορά – ίσως για αποσφαλμάτωση ή επέκτασή του – δεν θα θυμάστε τι εννοούσατε όταν το γράψατε!

Παρόλο που εδώ το όνομα που δόθηκε στη μεταβλητή επεξηγεί επαρκώς το τι περιέχει, προσθέστε κάποιο σχόλιο κοντά στη δήλωσή της, το οποίο να εξηγεί ότι η μεταβλητή περιέχει κάποια ηλικία. Δεν έχει σημασία πώς θα το γράψετε, αρκεί να είναι κατανοητό! Προσέξτε να διατηρήσετε το πρόγραμμα ευανάγνωστο.

Στη συνέχεια, θα μάθετε πώς να δώσετε τιμή στη μεταβλητή από το πληκτρολόγιο. Θυμηθείτε ότι γράφετε ένα πρόγραμμα, το οποίο θα μεταφραστεί, ώστε να μπορεί να το χρησιμοποιήσει οποιοσδήποτε άλλος χρήστης. Άρα, την τιμή στη μεταβλητή θα την δίνει ο τυχαίος χρήστης του προγράμματος. Αν δεν επιθυμούσατε κάτι τέτοιο, θα δίνατε κάποια συγκεκριμένη τιμή μέσω ανάθεσης ή αρχικοποίησης. Όμως εδώ θέλουμε σκόπιμα η τιμή να λαμβάνεται από το πληκτρολόγιο. Πώς όμως θα ξέρει ο τυχαίος αυτός χρήστης τι να πληκτρολογήσει και πότε; Και αυτό είναι κάτι που εσείς θα το υποδείξετε στην οθόνη με κατάλληλη εντολή. Πηγαίνετε λοιπόν στη συμβολοσειρά που τροποποιήσατε στην προηγούμενη άσκηση, και γράψτε στη θέση της τη συμβολοσειρά `"Please, type your age: "`.

Αν μεταφράσετε και εκτελέσετε το πρόγραμμά σας τώρα, το μόνο που θα κάνει, θα είναι η εκτύπωση της πιο πάνω συμβολοσειράς. Για να προχωρήσετε, προσθέστε τη γραμμή:

```
scanf ("%d", &age);
```

αμέσως κάτω από την προηγούμενη εντολή εκτύπωσης, και με την ίδια στοίχιση. Αν δε χρησιμοποιήσατε το όνομα `age` στη δήλωση που γράψατε νωρίτερα, βάλτε στη θέση του το όνομα που χρησιμοποιήσατε. Δεν έχει σημασία το πόσους χαρακτήρες διαστήματος θα αφήσετε πριν και μετά το `'(`, πριν και μετά το `','`, πριν και μετά τον τελεστή `'&`, ή πριν και μετά το `)'`.

Με τη γραμμή αυτή καλείται η συνάρτηση *scanf()* της βιβλιοθήκης *stdio*, η οποία διαβάζει από το πληκτρολόγιο με διαμόρφωση που ορίζει η πρώτη παράμετρος της συνάρτησης. Η διαμόρφωση δίνεται με τη βοήθεια της συμβολοσειράς "%d", όπου το '%' υποδεικνύει ότι ακολουθεί χαρακτήρας διαμόρφωσης, και το 'd' ως τέτοιος χαρακτήρας υποδεικνύει ακέραια τιμή. Για κάθε διαμόρφωση που υπάρχει στη συμβολοσειρά – εδώ μία – πρέπει να ακολουθεί και η κατάλληλη θέση στην οποία θα αποθηκευτεί η τιμή που αναμένεται να πληκτρολογηθεί. Η δεύτερη παράμετρος είναι εδώ η διεύθυνση μνήμης της μεταβλητής όπου θα αποθηκευτεί η ηλικία που ζητήθηκε από το χρήστη. Το σύμβολο '&' είναι τελεστής που δίνει τη διεύθυνση μιας μεταβλητής και θα τον μελετήσουμε αναλυτικά σε άλλο εργαστήριο.

Για να καταλάβετε καλύτερα τη διαμόρφωση, προσθέστε κάτω από την εντολή εισόδου που μόλις γράψατε μια εντολή εξόδου – με κλήση της συνάρτησης *printf()* – που να εκτυπώνει την ακέραια τιμή που διαβάστηκε:

```
printf("Your age is %d!\n", age);
```

Μπορείτε να δείτε εύκολα πώς λειτουργεί η διαμόρφωση στην έξοδο. Με την εντολή αυτή εκτυπώνεται η συμβολοσειρά "Your age is ", ακολουθούμενη από την τιμή της μεταβλητής *age* (ή όποιας άλλης γράψατε στη θέση της), και στη συνέχεια από την υπόλοιπη συμβολοσειρά "!n". Προσέξτε ότι εδώ δεν υπάρχει ο τελεστής '&', μια που θέλουμε την τιμή της μεταβλητής και όχι τη διεύθυνσή της.

Μεταφράστε και εκτελέστε το πρόγραμμά σας, ώστε να επαληθεύσετε την είσοδο από το πληκτρολόγιο. Μόλις σας ζητηθεί, πληκτρολογήστε την ηλικία σας και πατήστε Enter. Αμέσως μετά θα εμφανιστεί το πιο πάνω μήνυμα. Κλείστε το παράθυρο κονσόλας για να συνεχίσετε.

Ολοκληρώνοντας, προσθέστε κάτω από την προηγούμενη και τη γραμμή:

```
printf("You were born in year %d or %d.\n", 2017-age, 2016-age);
```

Εδώ βλέπετε ότι ανάλογα με το πόσες διαμορφώσεις υπάρχουν στη συμβολοσειρά διαμόρφωσης, άλλες τόσες παράμετροι πρέπει να ακολουθούν στην κλήση της συνάρτησης. Επίσης, οι τιμές που δίνονται για εκτύπωση στη συνάρτηση *printf()* μπορούν να προκύπτουν μέσα από εκφράσεις. Μεταφράστε και εκτελέστε το τελικό πρόγραμμα.

Παρατηρήστε ότι το πρόγραμμά σας δέχεται ως είσοδο οποιονδήποτε ακέραιο αριθμό, ακόμα και αρνητικό. Θα μπορούσατε να το περιορίσετε να λειτουργεί με μη προσημασμένους ακέραιους, αν δηλώνατε τη μεταβλητή ως μη προσημασμένο ακέραιο (δηλαδή τύπου *unsigned*), και ταυτόχρονα αλλάζατε το χαρακτήρα διαμόρφωσης των εντολών εισόδου/εξόδου από 'd' σε 'u'. Αν θέλετε, δοκιμάστε να κάνετε αυτή την αλλαγή, και δώστε για πείραμα αρνητικό αριθμό ως είσοδο!

Κλείστε το παράθυρο κονσόλας.

Αποθηκεύστε το πρόγραμμά σας και κλείστε το *Code::Blocks*.<sup>2</sup>

### Άσκηση 2: «Σύγκριση δύο ακεραίων»

Εκκινήστε το πρόγραμμα *Code::Blocks*. Στη συνέχεια δημιουργήστε μια καινούργια εργασία, ονομάστε την *lab1* και ορίστε τη θέση της όπως κάνατε στο προηγούμενο εργαστήριο. Θα προκύψει πάλι το γνωστό σας πρόγραμμα. Αυτή τη φορά, επιλέξτε από το μενού File την επιλογή «Save file as...», ώστε να δώσετε άλλο όνομα στο αρχείο

---

<sup>2</sup> Η χρήση της συνάρτησης *getchar()* για να παγώνετε την κονσόλα και να βλέπετε το αποτέλεσμα της εκτέλεσης θα πρέπει να γίνεται σε όλες τις ασκήσεις του εργαστηρίου, άσχετα αν δεν θα αναφέρεται.

του προγράμματος. Δώστε το όνομα *lab1.2.c* και πατήστε Enter. Στη συνέχεια πηγαίνετε στο μενού Project, επιλέξτε «Add files...» και κάντε κλικ στο όνομα του αρχείου που φτιάξατε. Πατήστε Enter για τη συνέχεια. Τώρα το παράθυρο επεξεργασίας κειμένου έχει ανοικτό το νέο αρχείο.

Μπορείτε αν θέλετε να διαγράψετε το αρχείο *main.c* μέσα από το *Code::Blocks*. Αν στο παράθυρο διαχείρισης εργασιών κάνετε κλικ πάνω στο όνομα του αρχείου για να το επιλέξετε και μετά πατήσετε Delete δεν θα διαγραφεί το αρχείο, αλλά απλά θα αποσυνδεθεί το όνομά του από την εργασία. Για να το διαγράψετε κανονικά, θα πρέπει να κάνετε κλικ στο Files του παράθυρου διαχείρισης, ώστε αφού εντοπίσετε το αρχείο μέσα στο σύστημα αρχείων, να το επιλέξετε με το ποντίκι, και μετά να το διαγράψετε με το πλήκτρο Delete και επακόλουθη επιβεβαίωση. Αν κάνατε τη διαγραφή, κάντε κλικ στο Projects του παράθυρου διαχείρισης, ώστε να επανέλθετε εκεί που ήσαστε. Αν δεν το κάνατε πιο πριν, αποσυνδέστε το όνομα *main.c* από την εργασία.

Δηλώστε τώρα δύο καθολικές ακέραιες μεταβλητές τύπου *int*, έστω *i1* και *i2*. Στη συνέχεια χρησιμοποιήστε τις συναρτήσεις *printf()* και *scanf()* με τον τρόπο που μάθατε στην προηγούμενη άσκηση, ώστε αντί της απλής εκτύπωσης του μηνύματος «Hello world!», στην αρχή της συνάρτησης *main()* να διαβάσετε από το πληκτρολόγιο δύο ακεραίους, υποδεικνύοντας πιο πριν με κατάλληλο μήνυμα στο χρήστη το πού και πότε να τους πληκτρολογήσει. Μπορείτε να επιβεβαιώσετε την ανάγνωση των δύο αριθμών, γράφοντας την εντολή:

```
printf("You typed the numbers %d and %d.\n", i1, i2);
```

Όπως και στην προηγούμενη άσκηση, η συνάρτηση *printf()* έχει δύο διαμορφώσεις για να εκτυπώσει στην οθόνη δύο ακεραίους αριθμούς μέσα από ισάριθμες παραμέτρους. Στην πραγματικότητα τόσο η *printf()* όσο και η *scanf()* μπορούν να έχουν μη ορισμένο πλήθος διαμορφώσεων, που όλες εμπεριέχονται σε μία και μόνο συμβολοσειρά διαμόρφωσης. Οι παράμετροι που ακολουθούν τη συμβολοσειρά διαμόρφωσης αντιστοιχούνται με τη σειρά σε αυτές τις διαμορφώσεις.

Στη συνέχεια σας ζητείται να γράψετε τον κώδικα που συγκρίνει τους δύο αριθμούς για να βρείτε το μικρότερο, και να τον εκτυπώσετε με κατάλληλο μήνυμα. Αυτό σημαίνει ότι πρέπει να χρησιμοποιήσετε κάποιον τελεστή σύγκρισης, πχ. τον '<' που εδώ μοιάζει να είναι ο πιο κατάλληλος. Η σύγκριση μεταξύ των *i1* και *i2* θα γίνει μέσα από τη λογική έκφραση της συνθήκης κάποιας εντολής *if*. Αν η συνθήκη είναι αληθής, θα πρέπει να εκτυπώνεται αντίστοιχο μήνυμα. Αλλιώς, με χρήση της επιλογής *else* θα εκτυπώνεται άλλο αντίστοιχο μήνυμα. Το ζητούμενο μήνυμα θα πρέπει να είναι κάτι σαν «The smallest of the two numbers is -3.», όπου αντί για -3 θα εκτυπώνεται ο μικρότερος από τους δύο αριθμούς που δόθηκαν. Παρατηρήστε ότι αν οι δύο αριθμοί είναι ίσοι, τότε όποιος και να γραφτεί θα είναι ο μικρότερος.

Μεταφράστε και εκτελέστε το πρόγραμμά σας. Αν χρειαστεί, διορθώστε τα λάθη σας και επαναλάβετε!

Τροποποιήστε τώρα το πρόγραμμα που γράψατε, ώστε πριν τη σύγκριση για μικρότερο, να ελέγχετε πρώτα για ισότητα με τη βοήθεια του τελεστή '==' και να εκτυπώνετε κατάλληλο μήνυμα. Χρησιμοποιήστε τη σύνθετη εντολή της C – δηλαδή μια σειρά εντολών μέσα σε άγκιστρα, ώστε μετά την εκτύπωση να γράψετε την εντολή επιστροφής «return 0;». Έτσι ο κώδικας θα τερματίζεται εδώ, αν η συνθήκη ισότητας είναι αληθής. Μετά τη σύνθετη εντολή που εισάγατε μπορεί να ακολουθεί απ' ευθείας ο κώδικας που γράψατε νωρίτερα. Δε χρειάζεται να ακολουθεί επιλογή *else*, αφού η εντολή επιστροφής διακόπτει την εκτέλεση της συνάρτησης.

Ξαναμεταφράστε και εκτελέστε το πρόγραμμά σας.

Αποθηκεύστε το αρχείο lab1.2.c και συνεχίστε με την επόμενη άσκηση. Για ασφάλεια, πριν συνεχίσετε, συνιστάται να αντιγράψετε το αρχείο αυτό και σε άλλη θέση στο δίσκο του υπολογιστή ή σε φορητή συσκευή αποθήκευσης.

### Άσκηση 3: «Σύγκριση και ταξινόμηση τεσσάρων ακεραίων»

Στην άσκηση αυτή σας ζητείται να υλοποιήσετε τον κώδικα που συγκρίνει μεταξύ τους τέσσερις ακεραίους αριθμούς που λαμβάνονται από το πληκτρολόγιο, εκτυπώνοντας στην οθόνη ως αποτέλεσμα τους αριθμούς ταξινομημένους σε αύξουσα σειρά. Μπορείτε να χρησιμοποιήσετε την προηγούμενη άσκηση ή να ξεκινήσετε κάποια νέα εργασία. Στην πρώτη περίπτωση, όπως και πριν, αποθηκεύστε το αρχείο με νέο όνομα *lab1.3.c*, ώστε να μη χάσετε το αρχείο της προηγούμενης άσκησης. Τώρα όμως *μη* διαγράψετε το προηγούμενο αρχείο. Στη δεύτερη περίπτωση, μπορείτε να ακολουθήσετε τα αρχικά βήματα της πρώτης άσκησης, δίνοντας όμως στο αρχείο το νέο όνομα *lab1.3.c*.

Γράψτε τον κώδικα – όμοιο με αυτόν που γράψατε νωρίτερα, για να δηλώσετε τέσσερις καθολικές μεταβλητές τύπου `int` και να εισάγετε σε αυτές από το πληκτρολόγιο τέσσερις ακεραίους αριθμούς. Αν δεν δουλεύετε πάνω σε αντίγραφο του προηγούμενου αρχείου, μπορείτε να έχετε ανοικτά και τα δύο αρχεία στο παράθυρο επεξεργασίας, και να αντιγράψετε με γνωστές λειτουργίες αντιγραφής και επικόλλησης (`copy – paste`) τα κομμάτια του αρχείου *lab1.2.c* που θα χρειαστείτε στο νέο αρχείο. Σε τέτοια περίπτωση, κάντε κλικ στο όνομα αρχείου, για να μεταβαίνετε στο αρχείο που θέλετε κάθε φορά να επεξεργάζεστε. Προσθέστε και μια γραμμή κλήσης της `printf()` για να εκτυπώνει ως επιβεβαίωση τους τέσσερις αριθμούς που διαβάστηκαν.

Για τη σύγκριση πολλών αριθμών μεταξύ τους επιβάλλεται η χρήση μιας επαναληπτικής δομής βρόχου. Όμως σε αυτό το εργαστήριο *δεν* θα χρησιμοποιήσετε τέτοιες δομές. Για τέσσερις αριθμούς, μια μη επαναληπτική υλοποίηση ταξινόμησης *δεν* είναι απαγορευτική σε πολυπλοκότητα κώδικα, αλλά θα καταλάβετε μόνοι σας ότι θα ήταν δύσκολο να υλοποιήσετε ταξινόμηση περισσότερων αριθμών με τέτοιο τρόπο.

Ο τρόπος με τον οποίο θα προχωρήσετε στην υλοποίηση αυτή είναι οι φωλιασμένες εντολές `if`. Έτσι, μετά την ανάγνωση και επιβεβαίωση ανάγνωσης, θα συγκρίνετε δύο από τους αριθμούς και θα λάβετε υποπεριπτώσεις για κάθε περίπτωση αληθούς και ψευδούς συνθήκης – χρησιμοποιώντας όπου χρειάζεται επιλογή `else`, αναλύοντας κάθε περίπτωση σε υποπεριπτώσεις σύγκρισης άλλων δύο αριθμών, *κοκ*, έως ότου να έχετε εξαντλήσει όλες τις πιθανές συγκρίσεις που απαιτούνται για να βρεθεί η σωστή διάταξη των αριθμών. Να προσέξετε ιδιαίτερα τη χρήση σύνθετων εντολών έναντι απλών, ώστε να μη χαθείτε στις επιλογές και μπείτε κατά λάθος από μία υποπερίπτωση σε άλλη! Γενικά συνιστάται να δημιουργήσετε σύνθετες εντολές (α) σε όσες υποπεριπτώσεις βάλετε πάνω από μία εντολή, και (β) σε όσες υποπεριπτώσεις έχετε μία εντολή, η οποία όμως είναι εντολή `if`, ώστε να είστε σίγουροι ότι οι επιλογές `else` που γράφετε ταιριάζουν με τις σωστές εντολές `if`. Σε κάθε τελική διάταξη θα πρέπει να βάλετε μια εντολή εκτύπωσης της σειράς των αριθμών.

Οι πρώτες γραμμές μιας πιθανής μορφής κώδικα που υλοποιεί την ταξινόμηση με αυτόν τον τρόπο θα είναι:

```
if (i1 < i2) {
    if (i2 < i3) {
        if (i3 < i4)
```

```
    printf("The correct order is: %d,%d,%d,%d\n",
           i1, i2, i3, i4);
else if (i2 < i4)
    printf("The correct order is: %d,%d,%d,%d\n",
           i1, i2, i4, i3);
else if ...
```

...

όπου βέβαια καλείστε να συμπληρώσετε τον υπόλοιπο κώδικα με όλους τους απαραίτητους συνδυασμούς αριθμών προς σύγκριση.

Όταν ολοκληρώσετε τον κώδικα, μεταφράστε και εκτελέστε το πρόγραμμα. Μην απογοητευτείτε αν δεν επιτύχει με την πρώτη, μια που συνήθως έτσι συμβαίνει σε αρχάριους προγραμματιστές!

Αποθηκεύστε το αρχείο του προγράμματος και μην ξεχάσετε να αντιγράψετε το αρχείο σε ασφαλή αποθήκευση.

### *Υποβολή ασκήσεων*

Για την υποβολή των δύο προγραμμάτων των ασκήσεων, να δημιουργήσετε ένα φάκελο με το όνομα και τον ΑΜ σας, πχ. georgios-dimitriou-1234, και να αντιγράψετε σε αυτόν τα αρχεία lab1.2.c και lab1.3.c. Κάντε δεξί κλικ με το ποντίκι πάνω από το φάκελο, και επιλέξτε «Compress and email...». Θα βγει αυτόματα η αποστολή ηλεκτρονικού ταχυδρομείου, όπου θα δώσετε τη διεύθυνση [dimitriu@uth.gr](mailto:dimitriu@uth.gr). Ως θέμα θα πρέπει υποχρεωτικά να βάλετε την ένδειξη *lab1-askhseis*. Κοινοποιήστε το μήνυμα και στον εαυτό σας για να βεβαιωθείτε ότι οι ασκήσεις του εργαστηρίου υποβλήθηκαν σωστά.

Αν δεν υπάρχει η επιλογή «Compress and email...» στον υπολογιστή σας, να εκκινήσετε την εφαρμογή ηλεκτρονικού ταχυδρομείου, να δημιουργήσετε ένα νέο μήνυμα στο οποίο να επισυνάψετε το φάκελο που φτιάξατε, είτε με είτε χωρίς συμπίεση, και να στείλετε το μήνυμα όπως περιγράφεται πιο πάνω.