ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Τμήμα Πληροφορικής και Τηλεπικοινωνιών • Εισαγωγή στον Προγραμματισμό

Περιεχόμενα

Κεφάλαιο 12: Καλές πρακτικές, συχνά λάθη και βιβλιογραφία

Ένα πρόγραμμα C πρέπει να είναι ...

- ... **σωστό**. Προφανώς! Πρόγραμμα που δίνει λάθος αποτελέσματα πρέπει να διορθωθεί.
- ... **αποδοτικό**. Τα προγράμματα πρέπει να δίνουν τα αποτελέσματά τους μέσα σε εύλογο χρονικό διάστημα. Βέβαια, ο ορισμός του "εύλογου" ποικίλει ανάλογα με την περίπτωση και τις συνθήκες. Σε κάθε περίπτωση, θα πρέπει να προσπαθούμε το πρόγραμμά μας να έχει τη μέγιστη δυνατή απόδοση. Επίσης, θα πρέπει να ελέγχουμε τη συμπεριφορά του και για, πιθανώς ασυνήθιστα, μεγάλες εισόδους. Πάντως, ένα πρόγραμμα που θεωρητικά κάποια στιγμή θα τερματίσει, αλλά μπορεί να μην ζούμε για να το δούμε, δεν έχει πρακτική αξία.
- ... **εύρωστο**. Δεν πρέπει το πρόγραμμα να βγάζει λάθη εκτέλεσης, οποιαδήποτε και αν είναι η είσοδος που δέχτηκε. Δεν αρκεί να συμπεριφέρεται ομαλά μόνο για τις αναμενόμενες εισόδους.
- ... **τεκμηριωμένο**. Η τεκμηρίωση συνίσταται στην καλή επιλογή ονομάτων μεταβλητών, συναρτήσεων, κλπ., καθώς και στην προσθήκη σχολίων. Η τεκμηρίωση πρέπει να είναι τόση όση χρειάζεται για να κάνει το πρόγραμμα κατανοητό. Ούτε λιγότερη, αλλά ούτε και περισσότερη.
- ... **ευανάγνωστο**. Πρέπει να έχει συνεπή στοίχιση, να αποφεύγονται γραμμές με περισσότερους από 80 χαρακτήρες, να υπάρχουν κενά γύρω από τελεστές, όπου αυτό

βελτιώνει την αναγνωσιμότητα.

Συχνά προγραμματιστικά λάθη στην C

- Να κάνουμε διαίρεση μεταξύ ακεραίων, ευελπιστώντας ότι θα πάρουμε σαν αποτέλεσμα έναν αριθμό κινητής υποδιαστολής, ενώ στην πραγματικότητα παίρνουμε το ακέραιο πηλίκο της διαίρεσης.
- Να νομίσουμε ότι σε μία ακέραια μεταβλητή μπορούμε να φυλάξουμε τιμή οσοδήποτε μεγάλη, ή ότι σε μία μεταβλητή κινητής υποδιαστολής μπορούμε να έχουμε όση ακρίβεια θέλουμε.
- Να κάνουμε λάθος στην οριακή συνθήκη τερματισμού μίας δομής επανάληψης.
- Να χρησιμοποιήσουμε για τελεστή σύγκρισης μέσα σε συνθήκη το `=`, αντί για το `==`.
- Να βάλουμε μετά από μία `for` ή μία `while` ένα `;` πριν την εντολή ή το μπλοκ εντολών που αποτελούν το σώμα τους.
- Να ξεχάσουμε ότι τα στοιχεία ενός πίνακα με διάσταση `N` αριθμούνται από το `0` έως το `N-1`, και όχι από το `1` έως το `N`.
- Να χειριστούμε απρόσεκτα τους δείκτες, για παράδειγμα να δοκιμάσουμε να γράψουμε εκεί που πιστεύουμε ότι δείχνει ένας δείκτης, χωρίς να έχουμε κάνει την απαιτούμενη δέσμευση μνήμης.
- Να δεσμεύουμε δυναμικά μνήμη με επαναληπτικό τρόπο και να μην την αποδεσμεύουμε όταν δεν την χρειαζόμαστε.
- Να βάλουμε `;` στο τέλος μίας οδηγίας `#define`.
- Να ξεχάσουμε να βάλουμε παρενθέσεις γύρω από το κείμενο αντικατάστασης μίας μακροεντολής, που ορίζουμε με `#define`, και όπου αλλού χρειάζεται μέσα σ' αυτό.
- Να παρεξηγήσουμε τις προτεραιότητες των τελεστών, για παράδειγμα να γράψουμε

  ```
  while (ch = getchar() != EOF)
  ```

  αντί για

  ```
  while ((ch = getchar()) != EOF)
  ```

- Να κάνουμε σύγχυση ανάμεσα στην έννοια της προτεραιότητας των τελεστών, που έχει να κάνει με το πού εφαρμόζεται ένας τελεστής, με τη σειρά εκτέλεσης των ενεργειών που προσδιορίζουν οι τελεστές. Για παράδειγμα, το `*p++` είναι όντως το ίδιο με το `*(p++)`, το οποίο όμως πολλοί νομίζουν, λάθος, ότι σημαίνει ότι πρώτα θα αυξηθεί ο δείκτης `p` και μετά θα προσπελάσουμε το περιεχόμενο της νέας διεύθυνσης. **ΟΧΙ**. Απλώς, ο τελεστής `++` εδώ είναι μεταθεματικός, δηλώνοντας ότι πρώτα θα προσπελασθεί το περιεχόμενο της διεύθυνσης που δείχνει ο `p` και μετά θα αυξηθεί αυτός. Οι παρενθέσεις γύρω από το `p++`

δείχνουν πού εφαρμόζεται ο τελεστής `++` (στον δείκτη `p`) και όχι πότε θα εκτελεσθεί η πράξη που υποδεικνύει.

- Να χειριστούμε απρόσεκτα μεταβλητές με το ίδιο όνομα, παρεξηγώντας τις εμβέλειες καθεμιάς, και να χρησιμοποιούμε κάποια απ' αυτές, ενώ πραγματικά χρησιμοποιούμε κάποια άλλη.
- Να θεωρήσουμε ότι τοπικές μεταβλητές είναι αρχικοποιημένες (για παράδειγμα, σε `0`), παρότι εμείς δεν έχουμε κάνει ρητά κάτι τέτοιο.
- Να διατυπώσουμε απρόσεκτα εμφωλευμένες εντολές `if`, παραλείποντας άγκιστρα { και } σε περιπτώσεις που χρειάζονται, με αποτέλεσμα η εντολή να είναι μεν συντακτικά σωστή, οπότε ο μεταγλωττιστής δεν διαμαρτύρεται, αλλά να σημαίνει κάτι άλλο από αυτό που είχαμε πρόθεση να γράψουμε.
- Να ξεχάσουμε κρίσιμα `break` στις περιπτώσεις μίας εντολής `switch`.
- Να ταυτίσουμε το `'A'` με το `"A"`.
- Να παραβλέψουμε το γεγονός ότι μία συμβολοσειρά τερματίζει με τον χαρακτήρα `'\0'`.
- Να συγκρίνουμε συμβολοσειρές με τον τελεστή `==`, αντί με τη συνάρτηση `strcmp`.
- Να μην βάλουμε συνθήκη τερματισμού μίας αναδρομής.
- Να καλέσουμε την `scanf` περνώντας σ' αυτήν τις μεταβλητές που θέλουμε να διαβάσουμε, αντί για τις διευθύνσεις τους.

Βιβλιογραφία

- Brian W. Kernighan, Dennis M. Ritchie: "*Η Γλώσσα Προγραμματισμού C*", Prentice Hall (Ελληνική μετάφραση, εκδόσεις Κλειδάριθμος), 1988.
- Γ. Σ. Τσελίκης, Ν. Δ. Τσελίκας: "*C: Από τη Θεωρία στην Εφαρμογή*", 3η έκδοση, 2016.
- Νικόλαος Μισυρλής: "*Εισαγωγή στον Προγραμματισμό με την C*", 3η έκδοση, 2007.
- Νίκος Μ. Χατζηγιαννάκης: "*Η Γλώσσα C σε Βάθος*", 5η έκδοση, Εκδόσεις Κλειδάριθμος, 2017.
- Jeri R. Hanly, Elliot B. Koffman: "*Αρχές και Τεχνικές Προγραμματισμού με τη Γλώσσα C*", Pearson Education (Ελληνική μετάφραση, εκδόσεις Κριτική), 1η έκδοση, 2021.
- Δημήτριος Καρολίδης: "*Μαθαίνετε Εύκολα C*", 2η έκδοση, 2021.
- Herbert Schildt: "*Οδηγός της C*", 3η έκδοση, McGraw-Hill (Ελληνική μετάφραση, εκδόσεις Γκιούρδας), 2004.
- Eric S. Roberts: "*Η Τέχνη και Επιστήμη της C*", Addison-Wesley (Ελληνική μετάφραση, εκδόσεις Κλειδάριθμος), 2004.
- Brian W. Kernighan, Rob Pike: "*The Practice of Programming*", Addison-Wesley, 1999.
- Jon Bentley: "*Programming Pearls*", 2nd edition, Addison-Wesley, 2000.
- Steven S. Skiena: "*The Algorithm Design Manual*", Springer-Verlag, 1998.
- H. M. Deitel, P. J. Deitel: "*C: How to Program*", 3rd edition, Prentice Hall, 2001.

- Νικόλαος Θ. Αποστολάτος: "*Προγραμματισμός – Εφαρμογές*", 1995.
- Διομήδης Σπινέλλης: "*Ανάγνωση Κώδικα – Η Προοπτική του Ανοικτού Λογισμικού*", εκδόσεις Κλειδάριθμος, 2005.
- Brian W. Kernighan, Rob Pike: "*Το Περιβάλλον Προγραμματισμού Unix*", Prentice Hall (Ελληνική μετάφραση, εκδόσεις Κλειδάριθμος), 1989.
- Donald E. Knuth: "*The Art of Computer Programming: Sorting and Searching (Vol. 3)*", 2nd edition, Addison-Wesley, 1998.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: "*Introduction to Algorithms*", 2nd edition, The MIT Press, 2001.

Ευχαριστίες ...

- ... στην Επίκουρη Καθηγήτρια *Ιζαμπώ Καράλη* για τη μετάδοση πολύτιμης εμπειρίας, ιδεών και υλικού από το μάθημα του Αντικειμενοστραφούς Προγραμματισμού που διδάσκει στο Τμήμα μας από το ακαδημαϊκό έτος 2002–03.
- ... επίσης στην *Ιζαμπώ* και στους απόφοιτους του Τμήματος (προπτυχιακού και μεταπτυχιακού κύκλου) *Στέφανο Σταμάτη* και *Νίκο Ποθητό* που διάβαζαν με απεριόριστη υπομονή τα πρωτόλεια αυτών των διαφανειών/σημειώσεων, ενόσω γραφόντουσαν, και υπέδειξαν ένα πλήθος από λάθη και παραλείψεις που είχαν, κάνοντας ταυτόχρονα και προτάσεις για τη διόρθωσή τους.

Παναγιώτης Σταματόπουλος