

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ:

«ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ»

ΑΝΑΣΤΑΣΙΑΔΗΣ ΒΑΣΙΛΕΙΟΣ, ΑΜ: 123533

ΘΕ: ΠΛΣ-50

ΕΡΓΑΣΙΑ Ε1

ΕΙΣΑΓΩΓΙΚΗ ΠΑΡΕΝΘΕΣΗ:

Καθώς ακόμα δεν έχω κατασταλάξει σε κάποιο από τα 2 στυλ *bracketing* για τη συγγραφή του κώδικα, ακολούθησα την εξής στρατηγική:

-Στις πρώτες 3 ασκήσεις χρησιμοποιώ κυρίως το στυλ

```
int something(){  
    codecode;  
}
```

-Στην τελευταία άσκηση αντίθετα το στυλ

```
int something()  
{  
    codecode;  
}
```

Χωρίς να έχω καταλήξει, η εντύπωση μου είναι πως το πρώτο στυλ είναι πιο εύχρηστο και ευανάγνωστο όταν χρησιμοποιείται σωστά, αλλά και πιο δύσκολο να χρησιμοποιηθεί σωστά.

Θέμα 1

Στο πρώτο θέμα, ο χρήστης καλείται να δώσει 3 στοιχεία (2 αριθμούς και ένα σύμβολο) μέσω μιας `scanf` σε 3 μεταβλητές (`float a`, `char b`, `float c`).

Στη συνέχεια, μέσω μιας `switch` για τη μεταβλητή `b` (το σύμβολο) ελέγχεται αν είναι ένα από τα επιτρεπόμενα σύμβολα και γίνεται η αντίστοιχη πράξη. Στη συνέχεια το πρόγραμμα τερματίζει (καθώς δεν αναφέρεται η δυνατότητα συνέχειας στην εκφώνηση).

Στην περίπτωση λανθασμένου συμβόλου στο `b` (όχι όμως και στην περίπτωση εισαγωγής μη αριθμητικού χαρακτήρα στα `a`, `c`, όπου και το πρόγραμμα μπαίνει σε αέναη λούπα) το πρόγραμμα το αντιλαμβάνεται και ζητά από το χρήστη να ξαναπροσπαθήσει.

Άσκηση 2

Ο κώδικας της 2^{ης} άσκησης αρχίζει με τη δήλωση και υλοποίηση των 4 συναρτήσεων που υπολογίζουν την περίμετρο του τετραγώνου, παραλληλογράμμου, τριγώνου και κύκλου αντίστοιχα.

(Στην περίπτωση του τετραγώνου, οι πράξη είναι $x+x+x+x$ αντί για $4*x$ για ευκολότερη

κατανόηση. Ο compiler θα κάνει ακριβώς το ίδιο ότι και να γράψω, οπότε δεν τίθεται θέμα απόδοσης).

Στη main που ακολουθεί, μέσα σε μια while(1) - (δηλαδή αέναη λούπα – από αυτήν θα βγεί με **break** όταν ο χρήστης κάνει την αντίστοιχη επιλογή) – το πρόγραμμα δέχεται την επιλογή του χρήστη και καλεί την αντίστοιχη συνάρτηση υπολογισμού **μέσα στην printf που χρησιμοποιείται για την εκτύπωση** του αποτελέσματος.

Εδώ η επιλογή περίπτωσης γίνεται με **if/else if** αντί για **switch**.

Το πρώτο if ελέγχει αν ο χρήστης επιθυμεί έξοδο.

Το δεύτερο if ελέγχει αν η είσοδος δεν είναι κάποια από τις ορθές επιλογές, και σε αυτή την περίπτωση το πρόγραμμα συνεχίζει και ξαναρωτάει τυπώνοντας πρώτα το αντίστοιχο μήνυμα.

Ασκηση 3

Εδώ ο κώδικας είναι μικρότερος και σαφώς πιο εύκολο να τον ακολουθήσει κανείς. Και πάλι χρησιμοποιείται μια αέναη λούπα από την οποία το πρόγραμμα βγαίνει με **break** όταν ο χρήστης δώσει ως είσοδο το 0.

Στη συνέχεια, ο χρήστης δίνει τον (δεκαδικό) αριθμό του οποίου θέλει το ανεστραμμένο δυαδικό ανάπτγμα (ο αριθμός αυτός είναι ο x) και συμβαίνουν τα εξής:

- υπολογίζεται το $x \pmod{2}$

- αν είναι 1 τυπώνεται 1, αλλιώς τυπώνεται 0 (με τη χρήση του ?:)

- μπαίνει σε μια **do/while**, στην οποία σε κάθε βήμα διαιρεί το X με το 2 και επαναλαμβάνει τα παραπάνω 2 βήματα έως ότου γίνει το $x/2 == 0$

Τα πρώτα 2 βήματα γίνονται έξω από τη λούπα γιατί ήταν ένας εύκολος τρόπος να σταματάει η συνάρτηση στο σημείο που πρέπει (προσέξτε πως είναι σχεδόν μια ολοκληρωμένη λούπα, μόνο που λείπει το $x/2$ που γίνεται μόνο μέσα στη **do/while**).

Ασκηση 4

Για την τελευταία εργασία, επέλεξα (σε αντίθεση με τις προηγούμενες) να υλοποιήσω όλα τα printf στα Αγγλικά, καθώς έχει αρκετό κείμενο και δεν θα ήταν ευανάγνωστο με τη χρήση greeklish.

Ο κώδικας είναι εμφανώς περισσότερος σε σχέση με τα προηγούμενα θέματα, αλλά σχετικά ευθύς:

Δηλώνονται οι μεταβλητές που θα κρατήσουν το αρχείο εισόδου, τα γράμματα τις πινακίδας, τον αριθμό της πινακίδας και των αριθμό των θέσεων

```
FILE* fp,
```

```
char plateName[4],
```

int plateNum,

και int numberOfSeats

αντίστοιχα.

Αφού (μέσω fscanf) γεμίσουν οι παραπάνω μεταβλητές από το αρχείο γίνονται οι απαραίτητοι έλεγχοι όσον αφορά τις θέσεις του λεωφορείου:

- numberOfSeats<=5
- numberOfSeats>53
- (numberOfSeats - 5)%4 !=0 (για την ισχύ του τύπου $4N+5$)

Στη συνέχεια τυπώνονται (εφόσον όλα είναι σωστά) τα στοιχεία του λεωφορείου στην οθόνη, και αμέσως από κάτω το πρόγραμμα μπαίνει για άλλη μια φορά σε μια while(1) από την οποία βγαίνει με break μετά από την αντίστοιχη εντολή χρήστη.

Μέσα στη while, εμφανίζονται όλες οι επιλογές τις εκφώνησης (πλήρως υλοποιημένες) και στη συνέχεια με μια switch ελέγχεται η επιλογή και γίνονται οι αντίστοιχες πράξεις, Ο κώδικας σε συνδιασμό με τα σχόλια είναι εύκολο να κατανοηθεί.

ΤΕΛΟΣ ΠΡΩΤΗΣ ΕΡΓΑΣΙΑΣ

ΘΕΣΣΑΛΟΝΙΚΗ 9/11/2016