

## Σχόλια και παρατηρήσεις για το lab3.

### Μεταγλώττιση:

Ο κώδικας θα πρέπει να μεταγλωττίζεται χωρίς **warnings**.

"Μη ικανοποιητικό" σημαίνει ότι είχατε τουλάχιστον ένα warning.

Κώδικας που παράγει λάθη (errors) κατά τη μεταγλώττιση βαθμολογείται αυτομάτως με FAIL.

### Σχόλια:

Πρέπει να υπάρχουν γενικά σχόλια προγράμματος στην αρχή (ονοματεπώνυμο, ημερομηνία, σύντομη περιγραφή). Σχόλια μέσα στον κώδικα δεν είναι απολύτως απαραίτητα.

"Μέτριο" σημαίνει ότι τα σχόλια είναι ελλιπή ή ανακριβή.

"Μη ικανοποιητικό" σημαίνει ότι δεν υπάρχουν σχόλια.

### Στοιχισι & κενές γραμμές:

Όμοια με τις απαιτήσεις του **lab2**. Επίσης, καλό είναι να βάζετε ένα κενό πριν και μετά από τελεστές όπως το =, && κατά το ελάχιστο (για τον ίδιο λόγο που βάζετε κενά ανάμεσα στις λέξεις μιας πρότασης).

"Μέτρια" σημαίνει ότι η στοιχισι έχει ξεφύγει λίγο ή/και ότι υπάρχουν κάποιες κενές γραμμές αλλά όχι αρκετές.

"Μη ικανοποιητικά" σημαίνει ότι η στοιχισι είναι πολύ ασυνεπής ή/και δεν υπάρχουν αρκετές γραμμές ή είναι υπερβολικές (μεγάλα κενά ανάμεσα σε εντολές) ή/και οι σύνθετες εντολές (κυρίως στη δεύτερη άσκηση) δεν έχουν κανένα κενό πριν/μετά από τελεστές

### Τύποι και ονόματα μεταβλητών:

Οι τύποι των μεταβλητών θα πρέπει να είναι αυτοί που περιγράφονται στην εκφώνηση. Χρησιμοποιείτε `double` για αριθμούς κινητής υποδιαστολής για να έχουν μεγαλύτερη ακρίβεια οι υπολογισμοί. Τα ονόματα των μεταβλητών θα πρέπει να είναι σύντομα αλλά περιγραφικά.

"Μέτρια" ονόματα είναι `rad` (σημαίνει κάτι τελείως διαφορετικό από ακτίνα), `an_rad`, `Method` (δεν πρέπει να ξεκινά από κεφαλαίο).

"Μη ικανοποιητικά" είναι `d`, `g`, `A`, `B`.

### Χρήση `const`:

Κατά σύμβαση, `read-only`/σταθερές ποσότητες δηλώνονται με κεφαλαία γράμματα και περιέχουν τον προσδιοριστή `const` πριν από τον τύπο της μεταβλητής (π.χ. **`const double PI = 3.14159;`**). Περιμέναμε να είναι `const` το `π` στην πρώτη άσκηση και τουλάχιστον οι βασικές τιμές στη δεύτερη άσκηση.

"Μέτριο" : είναι μόνο το `π` στην πρώτη άσκηση ή έχουν όλες ονόματα με μικρά.

"Μη ικανοποιητικό": δεν υπάρχουν `const` ποσότητες σε καμία άσκηση.

### Είσοδος, ορθότητα πράξεων:

Στην πρώτη άσκηση η είσοδος δεν έχει κάτι ιδιαίτερο, αλλά στη δεύτερη πρέπει να προσέξετε στην ανάγνωση του χαρακτήρα. Ή πρέπει να βάλετε ένα κενό ανάμεσα στο " και στο `%c` για να καταναλωθεί το `enter` που ξέμεινε από την προηγούμενη `scanf`, ή να μη βάλετε κενό, αλλά να έχετε μια `getchar` πριν τη `scanf`. Η πρώτη λύση είναι προτιμότερη.

Στη δεύτερη άσκηση δώστε ιδιαίτερη προσοχή στη συνθήκη: Χρησιμοποιήστε λογικό τελεστή `&&` ή `||` (ανάλογα αν είστε Πέμπτη ή Παρασκευή) και όχι bitwise τελεστή `&` ή `|`. Προσοχή στη χρήση `==` (και όχι `=`) για σύγκριση ισότητας και στην κατάλληλη χρήση `<` ή `<=` ανάλογα με το τι θέλουμε να ελέγχουμε. Επίσης, καλό είναι να κάνετε χρήση παρενθέσεων αν δεν είστε σίγουροι για την προτεραιότητα των πράξεων.

Τέλος, ανάμεσα στους δύο παρακάτω τρόπους χρήσης του τελεστή ?: για να αναθέσουμε τιμή σε μια μεταβλητή είναι προτιμότερος ο δεύτερος γιατί είναι πιο “καθαρός” κώδικας, αποφεύγει εμφάνιση της ανάθεσης σε δύο σημεία, και αποφεύγει την πιθανότητα να γίνει συντακτικό λάθος λόγω έλλειψης ή κακής χρήσης παρενθέσεων γύρω από τις αναθέσεις.

"Μέτριο" για χρήση του πρώτου τρόπου ανάθεσης με ?:

"Μη ικανοποιητικό" για οποιοδήποτε λάθος στις πράξεις ή στη χρήση scanf

```
(weight_type == 'L' || price < 10)? shipping = LOW_CHARGE : shipping = HIGH_CHARGE;  
/* δεν προτιμάται αυτός ο τρόπος*/
```

```
shipping = (weight_type == 'L' || price < 10)? LOW_CHARGE : HIGH_CHARGE;
```

### Έξοδος προγράμματος:

Η έξοδος του προγράμματος πρέπει να είναι ΑΚΡΙΒΩΣ ίδια με την εκφώνηση. Επικεντρωνόμαστε στο format string των printf και όχι στο αν δε βγαίνουν σωστά τα νούμερα λόγω λάθος υπολογισμών νωρίτερα.

“Μέτριο” αν έχουν ξεφύγει κάποιοι λευκοί χαρακτήρες.

"Μη ικανοποιητικό" αν υπάρχουν πιο ουσιαστικές διαφορές (παράλειψη ' ', λάθος πλήθος δεκαδικών).