Σχόλια και παρατηρήσεις για το lab 4

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 πρέπει να βάζετε όταν δίνουν μια πληροφορία που δεν είναι προφανής. Για παράδειγμα, ένα σχόλιο που λέει /* ektypwsi apotelesmatos */ είναι κακό γιατί βλέπουμε την printf. Αν έχετε σχόλια που εξηγούν τα ονόματα των μεταβλητών τότε μάλλον έχετε άσχημα ονόματα μεταβλητών… Αλλά ένα σχόλιο δίπλα, για παράδειγμα, στις διάφορες ποσότητες της δεύτερης άσκησης που να λέει σε τι μονάδα είναι αυτά τα νούμερα είναι χρήσιμο γιατί δίνει μια επιπλέον πληροφορία που δε φαίνεται από τον κώδικα.

"Μέτριο" σημαίνει ότι τα σχόλια είναι ελλιπή ή ανακριβή ή υπερβολικά και μη χρήσιμα.

"Μη ικανοποιητικό" σημαίνει ότι δεν υπάρχουν σχόλια.

Στοίχιση & κενά: Ομοια με τις απαιτήσεις του **lab3**. Κοιτάμε στοίχιση, χρήση κενών γραμμών για διαχωρισμό "ενοτήτων" στο πρόγραμμα (αλλά όχι πάνω από μία διαδοχικά), χρήση χαρακτήρα space πριν και μετά από τελεστές (να μην είναι όλα κολλημένα σε κάθε έκφραση) και να μην εκτείνεται κώδικας ή σχόλια πέρα τις 80ής στήλης

"Μέτρια" σημαίνει ότι είναι κατά το πλείστον οκ με μικρές αποκλίσεις.

"Μη ικανοποιητικά" σημαίνει ότι η στοίχιση είναι πολύ ασυνεπής ή/και δεν υπάρχουν αρκετές γραμμές ή είναι υπερβολικές (μεγάλα κενά ανάμεσα σε εντολές)

Σταθερές: Κατά σύμβαση, read-only/σταθερές ποσότητες δηλώνονται με κεφαλαία γράμματα και περιέχουν τον προσδιοριστή const πριν από τον τύπο της μεταβλητής.

"Μέτριο" : υπάρχουν σταθερές, αλλά όχι για όλες τις σχετικές ποσότητες

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

Τύποι και ονόματα μεταβλητών/σταθερών: 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 θα πρέπει να είναι σύντομα αλλά περιγραφικά. Τα ονόματα const μεταβλητών θα πρέπει να είναι γραμμένα με κεφαλαία.

"Μέτρια" ονόματα είναι dist, hgt ή ονόματα const που είναι γραμμένα με μικρά.

"Μη ικανοποιητικά" είναι d, g.

Είσοδος: Προσοχή στην ανάγνωση του χαρακτήρα.

"Μη ικανοποιητικό" για οποιοδήποτε λάθος σε scanf.

if: Προσοχή στις συνθήκες και τη λογική του προγράμματος. Στην πρώτη άσκηση θέλουμε ο έλεγχος για τον πελάτη να λαμβάνει υπόψη και το 'y' και το 'Y'.

"Μη ικανοποιητικό" για οποιοδήποτε λάθος .

Τυπικά λάθη που πρέπει να προσέχετε:

- Στον έλεγχο του αν ένας χαρακτήρας είναι 'Y' ή 'y' είναι λάθος να γράψετε
  if (old_customer == 'Y' || 'y') Σύμφωνα με τους κανόνες προτεραιότητας της C, κατά τον υπολογισμό αυτής της έκφρασης γίνεται πρώτα η πράξη old_customer == 'Y' η οποία θα παράγει αληθές ή ψευδές (1 ή 0). Μετά αναλόγως με το προηγούμενο αποτέλεσμα γίνεται η πράξη 1 || 'y' ή η πράξη 0 || 'y' . Σε κάθε περίπτωση αυτό είναι αληθές γιατί το 'y' είναι μη-μηδενικό. Αρα, τελικά η συνθήκη είναι πάντα αληθής. Το σωστό βέβαια είναι
  if (old_customer == 'Y' || old_customer == 'y')
- Στον έλεγχο του αν η διαφορά χρόνου είναι μεταξύ -0.1 και 0.1 είναι λάθος να γράψετε
  if (time_difference > -0.1 || time_difference < 0.1)
  Η παραπάνω συνθήκη είναι αληθής για τιμές μεγαλύτερες του -0.1 ή μικρότερες του 0.1 Σκεφτείτε τι αποτέλεσμα θα βγει αν το time_difference είναι 5. Επειδή το 5 είναι μεγαλύτερο του του -0.1, η συνθήκη είναι αληθής, πράγμα που δε θέλετε. Το σωστό εδώ είναι να χρησιμοποιήσετε &&. Καλό είναι σε τέτοιες συνθήκες να κάνετε ένα γρήγορο έλεγχο με ενδεικτικές τιμές για αν σιγουρευτείτε ότι δεν έχετε λάθος τελεστή.
- Αρκετοί είχατε την εξής λογική:
  ```
  double discount;
  if (old_customer == 'Y' || old_customer == 'y')  {
      discount = 0.15*price;
  }
  price = price - discount;
  ```
  Παρατηρήστε πως το discount δεν έχει αρχικοποιηθεί. Αν είναι αληθής η συνθήκη της if, τότε το discount θα πάρει τιμή, αλλά είναι ψευδής, τότε το πρόγραμμα θα προχωρήσει στον υπολογισμό price-discount ενώ το discount περιέχει σκουπίδια! Επομένως το αποτέλεσμα θα είναι λάθος.

Πράξεις: Προσοχή στη δεύτερη άσκηση. Πρέπει να μετατρέψετε τα km/h σε m/sec. Εφόσον ένα χιλιόμετρο έχει 1000 μέτρα και μία ώρα έχει 60*60 δευτερόλεπτα, θα πρέπει να πολλαπλασιάσετε την ταχύτητα του μπιπ-μπιπ με 1000/(60*60). Προσοχή στο πώς θα γίνουν οι πράξεις. Αν κάνετε ακέραια διαίρεση, το παραπάνω παράγει 0. Πιθανές λύσεις:

- Να πολλαπλασιάσετε πρώτα την ταχύτητα (που είναι double) με το 1000 ώστε να βγει double αποτέλεσμα και μετά να διαιρέσετε με το 60*60, ή
- Να κάνετε typecast σε double τον παρονομαστή ή αριθμητή του κλάσματος, ή
- Αντί για 1000 να γράψετε 1000.0 ώστε να αντιμετωπιστεί ως double.

Επιπλέον, οι πράξεις της μετατροπής πρέπει να γίνουν από το πρόγραμμα και όχι να τις κάνετε σε κομπιουτεράκι και να χρησιμοποιήσετε το τελικό νούμερο, γιατί τότε δε θα βγάλετε ακριβή αποτελέσματα.

<u>Σημείωση για τη δεύτερη άσκηση</u>: Στον έλεγχο της διαφοράς χρόνου γίνεται δεκτή και η χρήση < και η χρήση <= λόγω ανακρίβειας στην εκφώνηση.

Εξοδος 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 (λάθος πλάτος ακεραίου, πλήθος δεκαδικών).