

### 1η Σειρά Εργαστηριακών Ασκήσεων

Οι εργαστηριακές ασκήσεις είναι **ατομικές**. Οι απαντήσεις θα πρέπει να υποβληθούν με **turnin**, το αργότερο μέχρι την **Τρίτη 22 Μαρτίου 2022, ώρα 23:59**.

Οι ώρες οι οποίες έχουν δεσμευτεί για το εργαστήριο του μαθήματος είναι την Παρασκευή 10-12πμ. Η παρουσία στο εργαστήριο τις παραπάνω ώρες δεν είναι υποχρεωτική. Μπορείτε να έρχεστε στο εργαστήριο τις ώρες αυτές για όποια βοήθεια χρειάζεστε σχετικά με την εκπόνηση των εργαστηριακών ασκήσεων και γενικότερα τον προγραμματισμό στη γλώσσα Haskell, καθώς και για την επίλυση προβλημάτων που παρουσιάζονται κατά τη συγγραφή των προγραμμάτων στο πλαίσιο των εργαστηριακών ασκήσεων. Για αντίστοιχα προβλήματα ή απορίες που θα προκύψουν στο διάστημα από την περάτωση του εργαστηρίου μέχρι την υποβολή της εργασίας μπορείτε να επικοινωνήσετε με την κ. Βίκυ Σταμάτη την Τρίτη 10:00-12:00 είτε δια ζώσης (Γραφείο B11) είτε μέσω ηλεκτρονικού ταχυδρομείου (vstamati@uoi.gr). Θα απαντηθούν μόνο ηλεκτρονικά μηνύματα που έχουν σταλεί από τον ιδρυματικό σας λογαριασμό.

Πριν ξεκινήσετε να γράφετε τα προγράμματα που ζητούνται στις ασκήσεις της σειράς αυτής, **διαβάστε πολύ προσεκτικά τις αναλυτικές οδηγίες** που ακολουθούν.

### Οδηγίες

- Για να εγκαταστήσετε τη Haskell στον υπολογιστή σας, μπορείτε να κατεβάσετε το διεργημένα hugs από το σύνδεσμο

<https://www.haskell.org/hugs/pages/downloading-May2006.htm>

Συνοπτικές οδηγίες για τη χρήση του hugs υπάρχουν στις σημειώσεις.

- Πριν ξεκινήσετε να γράφετε τα προγράμματα που ζητούνται στις παρακάτω ασκήσεις, θα ήταν χρήσιμο να γράψετε σε ένα αρχείο ορισμένες από τις συναρτήσεις των σημειώσεων, να φορτώσετε το αρχείο στον hugs και να αποτιμήσετε παραστάσεις που χρησιμοποιούν τις συναρτήσεις αυτές, έτσι ώστε να εξοικειωθείτε με την γλώσσα Haskell και το διεργημένα της.
- Για τη συγγραφή των προγραμμάτων επιτρέπεται να χρησιμοποιήσετε προκαθορισμένες συναρτήσεις και προκαθορισμένους τελεστές **μόνο εφόσον αναφέρονται στις σημειώσεις του μαθήματος**. Δεν επιτρέπεται η χρήση του import.
- Για τη συγγραφή των συναρτήσεων θα πρέπει να χρησιμοποιήσετε το αρχείο πρότυπο Lab1.hs (που υπάρχει στην ιστοσελίδα του μαθήματος), στο οποίο υπάρχουν έτοιμες οι δηλώσεις τύπων

των συναρτήσεων που θα πρέπει να κατασκευάσετε καθώς και μία ισότητα που ορίζει τις συναρτήσεις ώστε να επιστρέφουν μία προκαθορισμένη τιμή για όλες τις τιμές των ορισμάτων. Για να απαντήσετε σε μία άσκηση μπορείτε να αντικαταστήσετε την παραπάνω ισότητα με τις κατάλληλες ισότητες που ορίζουν την τιμή της συνάρτησης. **Δεν θα πρέπει να τροποποιήσετε το τύπο ούτε το όνομα της συνάρτησης.**

- Μπορείτε να χρησιμοποιήσετε όσες βοηθητικές συναρτήσεις θέλετε, οι οποίες θα καλούνται από τις συναρτήσεις που σας ζητείται να υλοποιήσετε. Σε καμία περίπτωση δεν θα πρέπει να προσθέσετε άλλα ορίσματα στις συναρτήσεις που σας ζητούνται (καθώς αυτό συνεπάγεται αλλαγή του τύπου τους).
- **Αν χρησιμοποιήσετε προκαθορισμένες συναρτήσεις ή τελεστές που δεν αναφέρονται στις σημειώσεις του μαθήματος ή αν χρησιμοποιήσετε το `import` για να ενσωματώσετε έτοιμο κώδικα, η αντίστοιχη άσκηση δεν θα βαθμολογηθεί.**
- Ο έλεγχος της ορθότητας των απαντήσεων θα γίνει με ημι-αυτόματο τρόπο. Σε καμία περίπτωση δεν θα πρέπει ο βαθμολογητής να χρειάζεται να κάνει παρεμβάσεις στο αρχείο που θα υποβάλετε. Συνεπώς θα πρέπει να λάβετε υπόψη τα παρακάτω:
  1. Κάθε μία από τις συναρτήσεις που σας ζητείται να υλοποιήσετε θα πρέπει να έχει το συγκεκριμένο όνομα και το συγκεκριμένο τύπο που περιγράφεται στην εκφώνηση της αντίστοιχης άσκησης και που υπάρχει στο αρχείο πρότυπο `Lab1.hs`. **Αν σε κάποια άσκηση το όνομα ή ο τύπος της συνάρτησης δεν συμφωνεί με αυτόν που δίνεται στην εκφώνηση, η άσκηση δεν θα βαθμολογηθεί.**
  2. Το αρχείο που θα παραδώσετε δεν θα πρέπει να περιέχει συντακτικά λάθη. Αν υπάρχουν τμήματα κώδικα που περιέχουν συντακτικά λάθη, τότε θα πρέπει να τα διορθώσετε ή να τα αφαιρέσετε πριν από την παράδοση. **Αν το αρχείο που θα υποβάλετε περιέχει συντακτικά λάθη, τότε ολόκληρη η εργαστηριακή άσκηση θα μηδενιστεί.**
  3. Οι συναρτήσεις θα πρέπει να επιστρέφουν αποτέλεσμα για όλες τις τιμές των ορισμάτων που δίνονται για έλεγχο στο τέλος κάθε άσκησης. Αν κάποιες από τις τιμές που επιστρέφουν οι συναρτήσεις δεν είναι σωστές, αυτό θα ληφθεί υπόψη στη βαθμολογία, ωστόσο η άσκηση θα βαθμολογηθεί κανονικά. **Αν ωστόσο οι συναρτήσεις δεν επιστρέφουν τιμές για κάποιες από τις τιμές ελέγχου (π.χ. προκαλούν υπερχείλιση στοίβας, ατέρμονο υπολογισμό ή κάποιο σφάλμα χρόνου εκτέλεσης) τότε η αντίστοιχη άσκηση δεν θα βαθμολογηθεί.**
  4. Κατα τη διόρθωση των ασκήσεων οι βαθμολογητές δεν θα κάνουν κλήσεις στις βοηθητικές συναρτήσεις που ενδεχομένως θα χρησιμοποιήσετε. Η χρήση των βοηθητικών συναρτήσεων θα πρέπει να γίνεται μέσα από τις συναρτήσεις που σας ζητείται να υλοποιήσετε.
- Μετά το τέλος της εκφώνησης κάθε άσκησης δίνονται τιμές που μπορείτε να χρησιμοποιήσετε για έλεγχο της ορθότητας των συναρτήσεων.
- Για υποβολή με turnin γράψτε:

**turnin Haskell-1@myy401 Lab1.hs**

### Άσκηση 1.

Μία μεγάλη αλυσίδα καταστημάτων εφαρμόζει μία πολιτική προσφορών και εκπτώσεων έτσι ώστε να αυξήσει τις πωλήσεις των προϊόντων της. Σύμφωνα με την πολιτική των πωλήσεων της αλυσίδας αυτής για την αγορά κάθε προϊόντος:

- στις πρώτες πέντε συσκευασίες του προϊόντος η μία είναι δωρεάν
- στις επόμενες τέσσερις συσκευασίες του προϊόντος η μία είναι δωρεάν
- σε κάθε επιπλέον τρεις συσκευασίες του προϊόντος η μία είναι δωρεάν.
- αν το συνολικό κόστος που προκύπτει από την αγορά του προϊόντος είναι μεγαλύτερο από 100 ΕΥΡΩ, τότε υπάρχει επιπλέον έκπτωση 10%.

Γράψτε μία συνάρτηση `amount` σε Haskell η οποία θα δέχεται ως ορίσματα το πλήθος των συσκευασιών ενός προϊόντος που αγοράζει κάποιος πελάτης και την τιμή της μίας συσκευασίας του προϊόντος και θα επιστρέφει το συνολικό κόστος που προκύπτει με βάση την πολιτική πωλήσεων της αλυσίδας που περιγράφεται παραπάνω. Ο τύπος της συνάρτησης θα πρέπει να είναι `Int->Float->Float`. Αν κάποιο από τα δύο ορίσματα δεν είναι θετικός αριθμός, τότε η συνάρτηση `amount` θα πρέπει να επιστρέφει την τιμή 0.

Για τη μετατροπή ακέραιου σε πραγματικό χρησιμοποιήστε τη συνάρτηση `fromIntegral`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> amount 3 32.0
96.0
Main> amount 4 32.0
115.2
Main> amount 5 32.0
115.2
Main> amount 8 0.15
1.05
Main> amount 9 0.15
1.05
Main> amount 35 4.25
95.625
Main> amount 36 4
100.0
Main> amount 37 4
93.6
Main> amount (-5) 15.55
0.0
Main> amount 10 (-3.77)
0.0
```

## Άσκηση 2.

Μία εταιρία κινητής τηλεφωνίας χρεώνει κάθε κλήση διάρκειας μέχρι 3 λεπτά προς οποιονδήποτε αριθμό με 0.58 ΕΥΡΩ και αν η διάρκεια της κλήσης υπερβεί τα 3 λεπτά, τότε ο επιπλέον χρόνος χρεώνεται με 0.003 ΕΥΡΩ το δευτερόλεπτο. Κλήσεις με μηδενική διάρκεια δεν χρεώνονται.

Γράψτε μία συνάρτηση `cost` σε Haskell η οποία θα δέχεται ως ορίσματα τις ώρες έναρξης και λήξης μίας κλήσης και θα υπολογίζει τη συνολική χρέωση για την κλήση. Η ώρα αναπαρίσταται ως μία τριάδα ακεραίων (για παράδειγμα η ώρα 15:18:31 αναπαρίσταται ως (15, 18, 31)). Ο τύπος της συνάρτησης θα πρέπει να είναι `(Int,Int,Int)->(Int,Int,Int)->Float`. Μπορείτε να υποθέσετε ότι τα δύο ορίσματα είναι πάντοτε έγκυρα (δηλαδή αντιστοιχούν σε σωστές ένδειξεις ώρας) και επιπλέον ότι η διάρκεια μίας κλήσης είναι μικρότερη από 24 ώρες.

Για τη μετατροπή ακεραίου σε πραγματικό χρησιμοποιήστε τη συνάρτηση `fromIntegral`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> cost (17,23,15) (17,23,15)
0.0
Main> cost (8,12,3) (8,12,58)
0.58
Main> cost (12,0,35) (12,1,24)
0.58
Main> cost (16,58,35) (17,0,0)
0.58
Main> cost (23,59,42) (0,1,40)
0.58
Main> cost (14,32,8) (14,35,17)
0.607
Main> cost (14,57,4) (15,0,23)
0.637
Main> cost (23,59,59) (0,2,59)
0.58
Main> cost (3,15,22) (11,55,8)
93.598
Main> cost (19,43,48) (1,5,7)
57.877
```