

4η Σειρά Εργαστηριακών Ασκήσεων

Οι εργαστηριακές ασκήσεις είναι ατομικές. Οι απαντήσεις θα πρέπει να υποβληθούν με **turnin**, το αργότερο μέχρι την **Παρασκευή 16 Απριλίου 2021, ώρα 16:45**. Πριν ξεκινήσετε να γράφεται τα προγράμματα που ζητούνται στις ασκήσεις της σειράς αυτής, **διαβάστε πολύ προσεκτικά τις αναλυτικές οδηγίες** που ακολουθούν.

- 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 `Lab4.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 `Lab4.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.
 - Μπορείτε να συνδέεστε στην ομάδα 'ΕΡΓΑΣΤΗΡΙΟ στις ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΜΟΥ' στο MsTeams τις ώρες του εργαστηρίου (Τρίτη 10-12), για την επίλυση αποριών και προβλημάτων που ενδέχεται να συναντήσετε 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 Παρασκευή 10:00-12:00 είτε μέσω MsTeams (προσωπικό μήνυμα) είτε μέσω ηλεκτρονικού ταχυδρομείου (vstamati@uoi.gr). Θα απαντηθούν μόνο ηλεκτρονικά μηνύματα που έχουν σταλεί από τον ιδρυματικό σας λογαριασμό.
 - Για υποβολή με turnin γράψτε:

turnin Haskell-4@myy401 Lab4.hs

Άσκηση 1.

Γράψτε μία συνάρτηση `move` σε Haskell, η οποία θα δέχεται ως ορίσματα μία λίστα s οποιουδήποτε τύπου που υποστηρίζει σύγκριση για ισότητα, ένα στοιχείο x ίδιου τύπου με τα στοιχεία της λίστας και έναν ακέραιο n και θα επιστρέφει τη λίστα που προκύπτει από την s με τον παρακάτω τρόπο:

- αν το x δεν περιέχεται στην s , τότε η επιστρεφόμενη λίστα είναι η s .
- αν το x περιέχεται στην s τότε
 - αν n είναι θετικός αριθμός, τότε η επιστρεφόμενη λίστα προκύπτει από την s με μετακίνηση της πρώτης εμφάνισης του στοιχείου x κατά n θέσεις προς τα δεξιά, ή στο τέλος της λίστας αν η πρώτη εμφάνιση του x στην s ακολουθείται από λιγότερα από n στοιχεία.
 - αν n είναι αρνητικός αριθμός, τότε η επιστρεφόμενη λίστα προκύπτει από την s με μετακίνηση της πρώτης εμφάνισης του στοιχείου x κατά $|n|$ θέσεις προς τα αριστερά, ή στην αρχή της λίστας αν στην s υπάρχουν λιγότερα από $|n|$ στοιχεία πριν από την πρώτη εμφάνιση του x .
 - αν $n = 0$, τότε η επιστρεφόμενη λίστα προκύπτει από την s με διαγραφή της πρώτης εμφάνισης του στοιχείου x .

Ο τύπος της συνάρτησης θα πρέπει να είναι `Eq u => [u]->u->Int->[u]`. Η συνάρτηση θα πρέπει να επιστρέφει αποτέλεσμα ακόμη και στην περίπτωση που το πρώτο της όρισμα είναι μία άπειρη λίστα η οποία περιέχει το στοιχείο x .

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> move [] 3 5
[]
Main> move [1,2,3,4] 0 4
[1,2,3,4]
Main> move [1,2,3,4,3,2,1] 3 3
[1,2,4,3,2,3,1]
Main> move [1,2,3,4,5,6,7,8] 6 7
[1,2,3,4,5,7,8,6]
Main> move "skin" 'k' 0
"sin"
Main> move "factor.bit." '.' (-2)
"fact.orbit."
Main> move "0000.1111.00" '.' (-10)
".00001111.00"
Main> move ["one", "two", "three"] "one" (-2)
["one","two","three"]
Main> move ["one", "two", "three"] "one" 1
["two","one","three"]
Main> move ["one", "two", "three"] "three" 5
["one","two","three"]
Main> move ["one", "two", "three"] "three" (-2)
["three","one","two"]
```

```
Main> move [True] True 0
[]
Main> move [True] True 3
[True]
Main> move [True] True (-1003)
[True]
Main> head (move [1..] 1 91)
2
Main> head (tail (move [1,5..] 4005 (-1000)))
4005
```

Άσκηση 2.

Γράψτε μία συνάρτηση υψηλότερης τάξης `combine` σε Haskell, η οποία θα δέχεται ως ορίσματα δύο λίστες $s = [a_1, a_2, \dots, a_k]$ και $t = [b_1, b_2, \dots, b_\ell]$ με στοιχεία τύπου u και v αντίστοιχα, δύο συναρτήσεις f και g τύπου $u \rightarrow v \rightarrow w$ και μία συνάρτηση h τύπου $\text{Int} \rightarrow \text{Bool}$ και θα επιστρέφει ως αποτέλεσμα τη λίστα $[c_1, c_2, \dots, c_{\min(k,\ell)}]$ με στοιχεία τύπου w , όπου

$$c_i = \begin{cases} f(a_i, b_i) & \text{αν } h(i) \\ g(a_i, b_i) & \text{αλλιώς} \end{cases}$$

Ο τύπος της συνάρτησης θα πρέπει να είναι $[u] \rightarrow [v] \rightarrow (u \rightarrow v \rightarrow w) \rightarrow (u \rightarrow v \rightarrow w) \rightarrow (\text{Int} \rightarrow \text{Bool}) \rightarrow [w]$.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> combine [5,4,3,2] [7,8,9,10] (*) (^) odd
[35,65536,27,1024]
```

```
Main> combine [2,2..] [1..20] (*) (^) (\n -> mod n 5 == 1)
[2,4,8,16,32,12,128,256,512,1024,22,4096,8192,16384,32768,32,131072,262144,524288,
1048576]
```

```
Main> combine ["summer","drops","black","white"] ["time","rain","board","snow"]
(++)(\x y -> y++x) odd
["summertime","raindrops","blackboard","snowwhite"]
```

```
Main> combine ["summer","drops","black","white","time","rain","board","snow"] [1..]
(\x y -> (x,0)) (\x y -> ("",y)) even
[("",1),("drops",0),("",3),("white",0),("",5),("rain",0),("",7),("snow",0)]
```