ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ: ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΑΚΑΔ. ΕΤΟΣ: 2019-20

2η Σειρά Εργαστηριακών Ασκήσεων

Οι εργαστηριακές ασκήσεις είναι ατομικές. Οι απαντήσεις θα πρέπει να υποβληθούν με **turnin**, το αργότερο μέχρι την **Δευτέρα 18 Μαΐου 2020**. 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 Lab2.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 Lab2.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 όποιο πρόβλημα συναντήσετε κατά τη συγγραφή των προγραμμάτων στο πλάισιο των εργαστηριακών ασκήσεων μπορείτε να στέλνετε email στην κ. Βίκυ Σταμάτη (vstamati@cs.uoi.gr) με θέμα "MYY401 - LAB-2".

- Για υποβολή με turnin γράψτε (υποθέτοντας ότι έχετε γράψει το πρόγραμμα στο αρχείο Lab2.hs):

turnin Haskell-2@myy401 Lab2.hs

Ασκηση 1.

Γράψτε μία συνάρτηση `nearest` σε Haskell, η οποία θα δέχεται ως ορίσματα μία λίστα ακεραίων αριθμών και έναν ακέραιο αριθμό n και θα επιστρέφει τη θέση της λίστας η οποία περιέχει το στοιχείο με την ελάχιστη απόσταση από το n. Ως απόσταση δύο αριθμών θεωρούμε την απόλυτη τιμή της διαφοράς τους. Αν η λίστα περιέχει πολλά στοιχεία που έχουν την ίδια απόσταση από το n, τότε η συνάρτηση θα πρέπει να επιστρέφει τη θέση του πρώτου από αυτά. Η τιμή της συνάρτησης δεν ορίζεται αν η λίστα είναι κενή. Ο τύπος της συνάρτησης `nearest` θα πρέπει να είναι `[Int]->Int->Int`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> nearest [0] 5
1

Main> nearest [2,7] 4
1

Main> nearest [2,7] 5
2

Main> nearest [16,25,39,42,20,50,64,72,10,48] 17
1

Main> nearest [16,25,39,42,20,50,64,72,10,48] 46
10

Main> nearest [16,25,39,42,20,50,64,72,10,48] 21
5

Main> nearest [16,25,39,42,20,50,64,72,10,48] 18
1

Main> nearest [16,25,39,42,20,50,64,72,10,48] 45
4
```

Ασκηση 2.

Γράψτε μία συνάρτηση `smooth` σε Haskell, η οποία θα δέχεται ως όρισμα μία λίστα ακεραίων αριθμών s και έναν θετικό ακέραιο αριθμό k και θα επιστρέφει τη λίστα ακεραίων αριθμών, το i-στό στοιχείο της οποίας είναι το ακέραιο μέρος του μέσου όρου των k στοιχείων που βρίσκονται στις θέσεις i, $i+1$, $\ldots$, $i+k-1$ της λίστας s. Η επιστρεφόμενη λίστα έχει $|s|-k+1$ στοιχεία, όπου $|s|$ το μήκος της λίστας s, αν $|s| \geq k$, αλλιώς είναι η κενή λίστα.

Για παράδειγμα, αν η s περιέχει 10 στοιχεία και $k=3$, τότε η επιστρεφόμενη λίστα περιέχει 8 στοιχεία. Το πρώτο στοιχείο της επιστρεφόμενης λίστας είναι το ακέραιο μέρος του μέσου όρου των τριών πρώτων στοιχείων της s. Το δεύτερο στοιχείο της επιστρεφόμενης λίστας είναι το ακέραιο μέρος του μέσου όρου του δεύτερου, του τρίτου και του τέταρτου στοιχείου της s. Το τελευταίο (όγδοο) στοιχείο της επιστρεφόμενης λίστας είναι το ακέραιο μέρος του μέσου όρου του όγδοου, του έννατου και του δέκατου στοιχείου της s.

Ο τύπος της συνάρτησης `smooth` θα πρέπει να είναι `[Int]->Int->[Int]`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> smooth [2,25,16,7,1,32,50,5,87,0] 1
[2,25,16,7,1,32,50,5,87,0]

Main> smooth [2,25,16,7,1,32,50,5,87,0] 2
[13,20,11,4,16,41,27,46,43]

Main> smooth [2,25,16,7,1,32,50,5,87,0] 3
[14,16,8,13,27,29,47,30]

Main> smooth [2,25,16,7,1,32,50,5,87,0] 5
[10,16,21,19,35,34]

Main> smooth [2,25,16,7,1,32,50,5,87,0] 8
[17,27,24]

Main> smooth [2,25,16,7,1,32,50,5,87,0] 12
[]

Main> smooth [101*i `mod` 113 | i<-[1..113]] 100
[57,56,55,55,55,56,56,56,57,57,57,56,55,54]
```

Ασκηση 3.

Έστω μία συμβολοσειρά η οποία αποτελείται από γράμματα του λατινικού αλφαβήτου και τον κενό χαρακτήρα. Ονομάζουμε λέξη κάθε ακολουθία συνεχόμενων γραμμάτων του λατινικού αλφαβήτου μέσα στη συμβολοσειρά, η οποία οριοθετείται στα αριστερά από τον κενό χαρακτήρα ή την αρχή της συμβολοσειράς και στα δεξιά από τον κενό χαρακτήρα ή το τέλος της συμβολοσειράς.

Για παράδειγμα στη συμβολοσειρά "o tempora o mores" υπάρχουν κατά σειρά οι λέξεις "o", "tempora", "o" και "mores".

Γράψτε μία συνάρτηση `swap` σε Haskell, η οποία θα δέχεται ως όρισμα μία συμβολοσειρά s και θα επιστρέφει τη συμβολοσειρά που προκύπτει αν στην s η πρώτη λέξη με τη δεύτερη αλλάξουν θέσεις μεταξύ τους, και το ίδιο γίνει για τις λέξεις στην τρίτη και την τέταρτη θέση, για τις λέξεις στην πέμπτη και την έκτη θέση, κοκ. Μπορείτε να υποθέσετε ότι η συμβολοσειρά περιέχει μόνο γράμματα του λατινικού αλφαβήτου και τον κενό χαρακτήρα, ότι δεν περιέχει συνεχόμενους κενούς χαρακτήρες και ότι δεν αρχίζει ούτε τελειώνει με τον κενό χαρακτήρα. Ο τύπος της συνάρτησης `swap` θα πρέπει να είναι `String->String`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> swap "haskell"
"haskell"

Main> swap "one two"
"two one"

Main> swap "you are smart"
"are you smart"

Main> swap "east west north south"
"west east south north"

Main> swap "abcdefghijklmnopqrstuvwxyz alphabet"
"alphabet abcdefghijklmnopqrstuvwxyz"

Main> swap "a b c d e f g h i j k l m n o p q r s t u v w x y z"
"b a d c f e h g j i l k n m p o r q t s v u x w z y"
```

Ασκηση 4.

Γράψτε μία συνάρτηση υψηλότερης τάξης `mapi` σε Haskell, η οποία θα δέχεται ως ορίσματα μία λίστα s τύπου `[u]` και μία συνάρτηση f τύπου `u->Int->v` και θα επιστρέφει μία λίστα τύπου `[v]`. Η επιστρεφόμενη λίστα θα πρέπει να έχει το ίδιο πλήθος στοιχείων μέ την s και το i-οστό στοιχείο της θα πρέπει να είναι το $f(s_i, i)$, όπου s_i είναι το i-οστό στοιχείο της s. Ο τύπος της συνάρτησης `mapi` θα πρέπει να είναι `[u]->(u->Int->v)->[v]`.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> mapi [90,80..10] (+)
[91,82,73,64,55,46,37,28,19]

Main> mapi [12,15,20,19,24,35,13,39,62,29] mod
[0,1,2,3,4,5,6,7,8,9]

Main> mapi [7,8,5,10,4,2,3,2,2,1] (^)
[7,64,125,10000,1024,64,2187,256,512,1]

Main> mapi ["island", "happiness", "stars", "surfing", "singing", "dancing"] (!!)
"spring"

Main> mapi "dream" (\x i -> toEnum (fromEnum x + i) :: Char)
"ether"

Main> mapi [[a,a+d..] | a<-[1,32,100,500], d<-[3,25,333,1040]] (!!)
[4,51,1000,4161,47,182,2363,8352,127,350,3763,12580,539,850,5495,17140]

Main> mapi [[a,a+d..] | d<-[1,32,100,500], a<-[3,25,333,1040]] (!!)
[4,27,336,1044,163,217,557,1296,903,1025,1433,2240,6503,7025,7833,9040]
```