



## Γλώσσες Προγραμματισμού II

Οι ασκήσεις πρέπει να παραδοθούν στους διδάσκοντες σε ηλεκτρονική μορφή μέσω του συνεργατικού συστήματος ηλεκτρονικής μάθησης [moodle.softlab.ntua.gr](http://moodle.softlab.ntua.gr). Η προθεσμία παράδοσης θα τηρείται αυστηρά. Έχετε δικαίωμα να καθυστερήσετε το πολύ μία άσκηση.

### Άσκηση 6 Γλώσσα Haskell

Προθεσμία παράδοσης: 11/2/2008

1. Να γραφεί σε Haskell η συνάρτηση υψηλής τάξης `zipWith`. Η `zipWith` έχει τύπο:

```
zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]
```

Η κλήση `zipWith f xs ys` εφαρμόζει την `f` στα αντίστοιχα στοιχεία των `xs` και `ys` και επιστρέφει το αποτέλεσμα. Για παράδειγμα:

```
zipWith (+) [1,2,3] [4,5] = [1+4,2+5] = [5,7]
```

2. Για να αποφασίσουμε εάν μια λίστα είναι ταξινομημένη, πρέπει να συγκρίνουμε κάθε στοιχείο με το επόμενο του και να ελέγξουμε εάν είναι μικρότερο ή ίσο του. Ένας τρόπος να κάνουμε το παραπάνω για τη λίστα `[a,b,c,d]` είναι ο εξής:
  - Αφαιρούμε το πρώτο στοιχείο της λίστας και κρατάμε την αρχική και την υπόλοιπη λίστα (`[b,c,d]`)
  - Συγκρίνουμε τα στοιχεία των δύο λιστών που έχουν αντίστοιχες θέσεις (π.χ. `a <= b`, `b <= c`, και `c <= d`), και παίρνουμε μια λίστα από boolean αποτελέσματα
  - Συνδυάζουμε όλα τα αποτελέσματα με χρήση του τελεστή `&&`
  - Το τελικό αποτέλεσμα είναι `True` εάν η αρχική λίστα είναι ταξινομημένη.

Να ορίσετε στη Haskell τη συνάρτηση

```
ordered :: Ord a => [a] -> Bool
```

με βάση την παραπάνω ιδέα, χρησιμοποιώντας τις συναρτήσεις υψηλής τάξης `zipWith` και `foldl`. Η συνάρτηση `ordered` σας δεν πρέπει να είναι αναδρομική.

3. Να γράψετε ένα πρόγραμμα Haskell που να μετατρέπει ένα αρχείο από ονόματα και διευθύνσεις ηλεκτρονικού ταχυδρομείου όπως το παρακάτω:

```
Kostis Sagonas          kostis@cs.ntua.gr
Nikolaos S. Papaspyrou  nickie@softlab.ntua.gr
```

σε μια ιστοσελίδα HTML όπως αυτή που δείχνουμε ως παράδειγμα:

```
<ul>
<li>Kostis Sagonas,
  <a href="mailto:kostis@cs.ntua.gr">
    kostis@cs.ntua.gr
  </a></li>
<li>Nikolaos S. Papaspyrou,
  <a href="mailto:nickie@softlab.ntua.gr">
    nickie@softlab.ntua.gr
  </a></li>
</ul>
```

**Σημείωση:** Στη γλώσσα HTML το `<ul>` tag εισάγει μια λίστα από αντικείμενα, το `<li>` tag εισάγει ένα στοιχείο αυτής της λίστας, και το `<a href=...>` εισάγει ένα σύνδεσμο. Μη δώσετε πολύ σημασία στη στοίχιση του HTML κώδικα που θα παράξετε. Με άλλα λόγια, δεν είναι απαραίτητο η στοίχισή του να είναι ακριβώς όπως η παραπάνω. Όμως καλό είναι κάποιος να μπορεί εύκολα να αποφασίσει με το μάτι εάν το αποτέλεσμα σας είναι σωστό.

Μπορείτε εύκολα να αναγνωρίσετε τι είναι email address διότι περιέχει το χαρακτήρα '@'. Το πρόγραμμά σας πρέπει να ορίζει μια συνάρτηση:

```
generateHtml :: IO ()
```

που να διαβάζει ονοματεπώνυμα και διευθύνσεις email από ένα αρχείο με όνομα "names", και να γράφει το αποτέλεσμα σε ένα αρχείο με όνομα "names.html".

## Άσκηση 7 Υλοποίηση γλωσσών συναρτησιακού προγραμματισμού

---

Προθεσμία παράδοσης: 11/2/2008

Στο παρακάτω URL υπάρχει μια πρόσφατη δημοσίευση που καταγράφει την ιστορία της ανάπτυξης της γλώσσας Haskell.

<http://research.microsoft.com/~simonpj/papers/history-of-haskell/history.pdf>

Διαβάστε το άρθρο, επικεντρώνοντας την προσοχή σας στα μέρη II και III, και αναφέρετε συνοπτικά (σε 2–4 σελίδες) τα σημεία που σας έκαναν μεγαλύτερη εντύπωση και γιατί.