ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ: ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΑΚΑΔ. ΕΤΟΣ: 2014-15
ΔΙΔΑΣΚΩΝ: Χ.ΝΟΜΙΚΟΣ

# 1η Σειρά Εργαστηριακών Ασκήσεων

Οι απαντήσεις θα πρέπει να το υποβληθούν με **turnin**, το αργότερο μέχρι την **Τρίτη 24 Μαρτίου 2015**, ώρα **20:00**.

- Για μεγαλύτερη ευκολία στην εκπόνηση αυτής και της επόμενης εργαστηριακής άσκησης σας συνιστώ:

  1. να δημιουργήσετε έναν κατάλογο Haskell κάτω από το home directory σας.
  2. να χρησιμοποιήτε αυτόν τον κατάλογο για αποθήκευση των αρχείων με τα προγράμματα Haskell που θα γράψετε.
  3. να μεταβαίνετε σε αυτόν τον κατάλογο πριν εκτελέσετε τον διερμηνέα hugs της Haskell.

- Αν θέλετε να χρησιμοποιήσετε τη Haskell στο δικό σας υπολογιστή, μπορείτε να κατεβάσετε το διερμηνέα hugs από το σύνδεσμο

  **https://www.haskell.org/hugs/**

  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 εξοικιωθείτε με την γλώσσα Haskell και το διερμηνέα της.

- Για τη συγγραφή των συναρτήσεων συνίσταται 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 την τιμή -2015 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).

- 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 αφαιρέσεται 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 αποτιμάτε τις τιμές score1 ... score5 που που ορίζονται στο αρχείο Lab1.hs για γρήγορο έλεγχο του πλήθους σωστών απαντήσεων σε κάθε άσκηση (ανάμεσα στις τιμές που δίνονται για έλεγχο στο τέλος της άσκησης) και του score για το συνολικό πλήθος σωστών απαντήσεων.

- Για υποβολή με turnin γράψτε (υποθέτωντας ότι έχετε γράψει το πρόγραμμα στο αρχείο Lab1.hs):

**turnin Haskell@myy401 Lab1.hs**

**Ασκηση 1**.
Σε ένα μάθημα που διδάσκετε σε κάποιο πανεπιστήμιο ο τελικός βαθμός προκύπτει από το βαθμό $a$ της τελικής εξέτασης, ο οποίος είναι ένας ακέραιος αριθμός μεταξύ 0 και 100 και το βαθμό του εργαστηρίου $b$, ο οποίος είναι ένας ακέραιος αριθμός μεταξύ 0 και 20.
Για τον υπολογισμό του τελικού βαθμού, ο οποίος είναι ένας ακέραιος αριθμός μεταξύ 0 και 100, πρώτα υπολογίζεται ο συνολικός βαθμός $c$, ο οποίος είναι το ακέραιο μέρος της παράστασης $\frac{8a}{10} + b$. Στη συνέχεια διακρίνονται οι παρακάτω περιπτώσεις:

- αν το $c$ είναι μεγαλύτερο του 47 και το $a$ δεν είναι μεγαλύτερο του 47, τότε ο τελικός βαθμός είναι 47.
- αν το $c$ και το $a$ είναι μεγαλύτερα του 47 και το $c$ είναι μικρότερο του 50, τότε ο τελικός βαθμός είναι 50.
- Σε κάθε άλλη περίπτωση ο τελικός βαθμός ισούται με το συνολικό βαθμό $c$.

Γράψτε μία συνάρτηση grade σε Haskell η οποία θα δέχεται ως ορίσματα τις βαθμολογίες της τελικής εξέτασης και του εργαστηρίου και θα επιστρέφει τον τελικό βαθμό. Ο τύπος της συνάρτησης θα πρέπει να είναι Int− >Int− >Int. Αν κάποιο από τα δύο ορίσματα έχει μή αποδεκτή τιμή, τότε η συνάρτηση grade θα πρέπει να επιστρέφει την τιμή -1.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> grade 100 20
100
Main> grade 100 0
80
Main> grade 0 20
20
Main> grade 80 20
84
Main> grade 59 10
57
Main> grade 48 10
50
Main> grade 48 9
47
Main> grade 40 20
47
Main> grade 20 100
-1
Main> grade 35 (-2)
-1
```

**Ασκηση 2**.
Ένας χώρος στάθμευσης αυτοκινήτων λειτουργεί από τις 6:00 το πρωί μέχρι τις 22:00 το βράδυ. Η στάθμευση μέχρι και τρεις ώρες χρεώνεται με 8 ευρώ. Για τις επόμενες τρεις ώρες (επιπλέον των τριών πρώτων) η χρέωση είναι 2 ευρώ ανά ώρα. Μετά τις έξι πρώτες ώρες η χρέωση είναι 1 ευρώ ανά ώρα. Η διάρκεια της σταθμευσης στογγυλοποιείται πάντοτε προς τα πάνω (για παράδειγμα πέντε ώρες και ένα λεπτό, χρεώνονται σαν έξι ώρες).
Γράψτε μία συνάρτηση parking σε Haskell η οποία θα δέχεται ως ορίσματα τις ώρες άφιξης και αναχώρησης ενός οχήματος και θα υπολογίζει τη συνολική χρέωση για τη στάθμευση. Η ώρα θα παριστάνεται ως ένα ζεύγος ακεραίων (για παράδειγμα η ώρα 10:15 παριστάνεται ως $(10, 15)$). Ο τύπος της συνάρτησης θα πρέπει να είναι (Int,Int)− >(Int,Int)− >Int. Μπορείτε να υποθέσετε ότι πάντοτε τα δύο ορίσματα παριστάνουν ώρες εντός του διαστήματος λειτουργίας του χώρου στάθμευσης και ότι η ώρα που παριστάνει το δεύτερο όρισμα είναι μεταγενέστερη αυτής που παριστάνει το πρώτο όρισμα.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> parking (13,59) (14,00)
8
Main> parking (15,30) (16,30)
8
Main> parking (8,45) (11,15)
8
Main> parking (6,15) (9,14)
8
Main> parking (12,22) (15,23)
10
Main> parking (16,15) (21,05)
12
Main> parking (10,55) (16,05)
14
Main> parking (6,05) (12,15)
15
Main> parking (8,32) (20,28)
20
Main> parking (6,00) (22,00)
24
```

**Ασκηση 3**.
Σε ένα τυχερό παιχνίδι κάθε Σάββατο γίνεται κλήρωση ενός 8-ψήφιου τυχερού αριθμού μεταξύ του 10000000 και του 99999999. Ένας παίκτης ο οποίος έχει στοιχηματίσει πάνω σε έναν αριθμό, κερδίζει ένα ποσό που εξαρτάται από το πλήθος των ψηφίων του αριθμού αυτού τα οποία ταυτίζονται με τα ψηφία του τυχερού αριθμού που βρίσκονται στις αντίστοιχες θέσεις, σύμφωνα με τον παρακάτω πίνακα:

| Ψηφία που συμφωνούν | Ποσό που κερδίζει ο παίκτης |
|---|---|
| 8 | 1000000 |
| 7 | 100000 |
| 6 | 8000 |
| 5 | 300 |
| 4 | 20 |
| 3 | 5 |
| 2 | 1 |
| λιγότερο από 2 | 0 |

Γράψτε μία συνάρτηση digits σε Haskell η οποία θα δέχεται ως ορίσματα τον τυχερό αριθμό και τον αριθμό που έχει επιλέξει ένα παίκτης και θα υπολογίζει το ποσό που κερδίζει ο παίκτης. Ο τύπος της συνάρτησης θα πρέπει να είναι Int− >Int− >Int. Μπορείτε να υποθέσετε ότι τα δύο ορίσματα είναι πάντοτε οκταψήφιοι αριθμοί.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> digits 13758455 13758455
1000000
Main> digits 22812509 22852509
100000
Main> digits 38954421 70954421
8000
Main> digits 42358921 42358002
300
Main> digits 12548706 14520786
20
Main> digits 80000012 71800009
5
Main> digits 42212553 22125531
1
Main> digits 58723493 83463738
0
Main> digits 78354623 46237835
0
Main> digits 34578364 83648364
20
```

**Ασκηση 4**.
Είναι γνωστό από τα μαθηματικά ότι αν $a, k, m$ είναι μη αρνητικοί ακέραιοι αριθμοί με $m \geq 2$ τότε το σύνολο $\{n \in \mathbb{N} \setminus \{0\} \mid (n+a)^k < m^n\}$ περιέχει άπειρα στοιχεία. Γράψτε μία συνάρτηση search σε Haskell η οποία θα δέχεται ως ορίσματα τρεις άκεραιους αριθμούς $a, k, m$ και θα επιστρέφει τον ελάχιστο θετικό ακέραιο $n$ για τον οποίο ισχύει $(n+a)^k < m^n$ (δηλαδή το ελάχιστο στοιχείο του συνόλου που περιγράφεται παραπάνω). Ο τύπος της συνάρτησης θα πρέπει να είναι Integer− >Integer− >Integer− >Integer. Μπορείτε να υποθέσετε ότι $a, k, m$ είναι μη αρνητικοί ακέραιοι αριθμοί και ότι $m \geq 2$.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> search 0 2 2
1
Main> search 1 2 2
6
Main> search 2 2 2
7
Main> search 2 5 2
24
Main> search 5 2 2
8
Main> search 1 10 3
32
Main> search 1 100 2
997
Main> search 1000 2 3
13
Main> search 100 100 100
117
Main> search 1000 1000 1000
1108
```

**Ασκηση 5**.
Γράψτε μία συνάρτηση sum2015 σε Haskell η οποία θα δέχεται ως ορίσματα δύο μη αρνητικούς άκεραιους αριθμούς $m, n$ και θα επιστρέφει το άθροισμα:

$$\sum_{i=m}^{n}(m+i)^n$$

Ο τύπος της συνάρτησης θα πρέπει να είναι Integer− >Integer− >Integer. Μπορείτε να υποθέσετε ότι $m, n$ είναι μη αρνητικοί ακέραιοι αριθμοί.

Για έλεγχο χρησιμοποιήστε τις παρακάτω τιμές:

```
Main> sum2015 0 1
1
Main> sum2015 0 2
5
Main> sum2015 0 3
36
Main> sum2015 0 10
14914341925
Main> sum2015 1 1
2
Main> sum2015 1 3
99
Main> sum2015 1 5
12200
Main> sum2015 3 4
3697
Main> sum2015 7 10
3981410573826
Main> sum2015 12 12
36520347436056576
```