ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ: ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
ΔΙΔΑΣΚΩΝ: Χ.ΝΟΜΙΚΟΣ

2η Σειρά Ασκήσεων

Οι παρακάτω ασκήσεις είναι για δική σας εξάσκηση και όχι για παράδοση. Οι λύσεις τους θα δοθούν αργότερα κατά τη διάρκεια του εξαμήνου.

Ασκηση 1.
Μετατρέψτε την παρακάτω συνάρτηση Python σε Haskell (για τη μετατροπή υποθέστε ότι οι παράμετροι και το αποτέλεσμα της p είναι ακέραιοι αριθμοί):

```
def p(a,b,c):
    while a>b or b>c:
        if a>b:
           a = (a+b) // 2
           c = c+1
        else: b=b-1
    return (a+b+c)
```

Ασκηση 2.
Γράψτε αναλυτικά πώς θα γίνει στη Haskell η αποτίμηση των παρακάτω παραστάσεων (όπου οι `power`, `comb` και `gcdEuc` ορίζονται όπως στις σημειώσεις):

- `power 4 6`
- `comb 10 3`
- `gcdEuc 52 91`

Ασκηση 3.
Έστω οι παρακάτω ορισμοί σε Haskell:

```
a :: Float
a = b-1

b :: Float
b = 3*a-1

f :: Float->Float->Float->Float->Float

f x y s t
    | x < 1.0 = (x+y) / (s*t)
    | y < 1.0 = f (x+y) (x-y) (t+s) (t-s)
    | otherwise = (x+y)^2
```

Δώστε όλα τα ενδιάμεσα βήματα του υπολογισμού που θα κάνει η Haskell για τη αποτίμηση της παράστασης `f 8.5 (f 0.7 1.3 20.0 0.20) a b`.

Ασκηση 4.
Έστω οι παρακάτω ορισμοί σε Haskell:

```
x :: Int
x = x `div` x
p :: Int -> Int -> Int
p m n
    | m == 0 = n
    | mod m 4 == 1 = p (m+2) (n-8)
    | otherwise = p (div m 4) (4*m)
```

Δώστε όλα τα ενδιάμεσα βήματα του υπολογισμού που θα κάνει η Haskell για τη αποτίμηση της παράστασης `p 5 x`.

Ασκηση 5.
Γράψτε αναλυτικά πώς θα γίνει στη Haskell η αποτίμηση των παρακάτω παραστάσεων (όπου οι `mergeInt`, `delete`, `insertInt`, `conc` ορίζονται όπως στις σημειώσεις):

- `mergeInt [5,16,20,24] [8,10,18]`
- `delete "white" ["green", "red", "blue", "white"]`
- `insertInt 12 [1,5,11,12,13,20]`
- `conc  [(1,4),(3,8),(5,5)] [(9,0),(0,3),(8,8)]`

Ασκηση 6.
Εξηγήστε γιατί οι παρακάτω παραλλαγές της `sqrtIntFast` δεν υπολογίζουν σωστά το ακέραιο μέρος της τετραγωνικής ρίζας ενός θετικού ακεραίου.

```
sq1 :: Int -> Int
sq1 n = sqrthlp 0 n
         where sqrthlp :: Int -> Int -> Int
               sqrthlp a b
                   | a==b
                         = a
                   | c*c > n
                         = sqrthlp a (c-1)
                   | otherwise
                          = sqrthlp c b
                     where c = (a+b) `div` 2


sq2 :: Int -> Int
sq2 n = sqrthlp 0 n
         where sqrthlp :: Int -> Int -> Int
               sqrthlp a b
                   | a==b
                         = a
                   | c*c > n
                         = sqrthlp a c
                   | otherwise
                          = sqrthlp (c+1) b
                     where c = (a+b) `div` 2
```

Ασκηση 7.
Έστω οι παρακάτω ορισμοί σε Haskell:

```
x :: Int
x = x `div` x

p :: Int -> [Int]
p n = x: n: p (n^n)
```

Γράψτε αναλυτικά πώς θα γίνει στη Haskell η αποτίμηση των παρακάτω παραστάσεων (όπου η `elemIntList` ορίζεται όπως στις σημειώσεις):

- `elemIntList 5 (p 3)`
- `elemIntList 4 (p 3)`