

a

i. Sum :: Int  $\rightarrow$  Integer  
 Sum n = ~~product~~ (take n Powers)  
 Sum

ii. lessThan :: ~~Integer  $\rightarrow$  Int~~ Integer  $\rightarrow$  Int  
 lessThan n = length (takeWhile (< n) powers)

iii. lessThan Sum :: ~~Int  $\rightarrow$  Integer~~ Integer  $\rightarrow$  Integer  
 lessThan Sum n = Sum (takeWhile (< n) powers)

iv. between :: Integer  $\rightarrow$  Integer  $\rightarrow$  [Integer]  
 between lo hi = takeWhile (<= hi) (dropWhile (< lo) Powers)

b. combinations :: [a]  $\rightarrow$  [b]  $\rightarrow$  [(a, b)]  
 combinations xs ys = [(x, y) | x <- xs, y <- ys]

c. capitalize :: String  $\rightarrow$  String (assumed input of Data.Chan)  
 capitalize w = unwords (map (\(x:xs) -> toUpper x : xs) (words w))

d. printLine :: (Int, String)  $\rightarrow$  IO ()

printLine (n, l) = ~~putStr~~ (

putStr (show n) >> putStr " " >> putStr l >> ~~putStr "\n"~~  
 >> putStr "\n"

printFile :: IO ()

printFile = do

input = readFile "in.txt"

let l = lines input

let lN = Zip [1..] l

mapM\_ printLine lN



## Question 2

page 2

a.  $\text{multiple} :: [a] \rightarrow \text{Bool}$

$\text{multiple} [] = \text{False}$

$\text{multiple} [x] = \text{False}$

$\text{multiple} xs = \text{True}$

b.  $\text{greater} :: \text{Ord } a \Rightarrow a \rightarrow [a] \rightarrow [a]$   
 $\text{greater } x \text{ } ys = \text{filter } (> x) \text{ } ys$

c.

i.  $[x+1 \mid x \leftarrow xs, x \leq 50]$

ii.  $[x \times 3 \mid x \leftarrow xs, x^2 < 100]$

d.  $\text{add Pairs} :: [\text{Int}] \rightarrow [\text{Int}]$

$\text{add Pairs} [] = []$

$\text{add Pairs} [x] = [x]$

$\text{add Pairs} (x_1 : x_2 : xs) = x_1 + x_2 : \text{add Pairs } xs$

e.  $\text{map Pairs} :: (a \rightarrow a \rightarrow a) \rightarrow [a] \rightarrow [a]$

$\text{map Pairs } f [] = []$

$\text{map Pairs } f [x] = [x]$

$\text{map Pairs } f (x_1 : x_2 : xs) = f \ x_1 \ x_2 : \text{map Pairs } f \ xs$



### Question 3

Page 3

a.

i Node 1 []

ii Node 1 [Node 2 [], Node 3 []]

iii Node 1 [Node 2 [Node 5 ~~1~~], Node 3 [], Node 4 []]

b.  $\text{root} :: \text{Tree } a \rightarrow a$   
 $\text{root} (\text{Node } x \_ ) = x$

c.  $\text{arity} :: \text{Tree } a \rightarrow \text{Int}$   
 $\text{arity} (\text{Node } \_ c) = \text{length } c$

d.  $\text{sumTree} :: \text{Tree } \text{Int} \rightarrow \text{Int}$   
 $\text{sumTree} (\text{Node } x []) = x$   
 $\text{sumTree} (\text{Node } x c) = x + \text{sum} (\text{map } \text{sumTree } c)$

e.  $\text{flipTree} :: \text{Tree } a \rightarrow \text{Tree } a$   
 $\text{flipTree} (\text{Node } x c) = \text{Node } x (\text{reverse} (\text{map } \text{flipTree } c))$

f.  $\text{mapTree} :: (a \rightarrow b) \rightarrow \text{Tree } a \rightarrow \text{Tree } b$   
 $\text{mapTree } f (\text{Node } x c) = \text{Node } (f x) (\text{map } (\text{mapTree } f) c)$