import Char
import Test.QuickCheck

-- Full credit is given for fully correct answers.
-- Partial credit may be given for partly correct answers.
-- Additional partial credit is given if there is indication of testing,
-- either using examples or quickcheck, as shown below.

-- 1a

c :: String -> Bool
c (w:ws) = isUpper w

f :: [String] -> String
f xs = concat [ x | x <- xs, c x ]

test1a =
f ["Once","Upon","a","Time"] == "OnceUponTime" &&
f ["no","capitals","!" ] == ""     &&
f ["ALL","CAPS"]      == "ALLCAPS"     &&
f ["ab","Cd","Ef","gh","ij"] == "CdEf"

-- 1b

g :: [String] -> String
g []        = ""
g (x:xs) | c x     = x ++ g xs
         | otherwise = g xs

test1b =
g ["Once","Upon","a","Time"] == "OnceUponTime" &&
g ["no","capitals","!" ] == ""     &&
g ["ALL","CAPS"]      == "ALLCAPS"     &&
g ["ab","Cd","Ef","gh","ij"] == "CdEf"

-- 1c

h :: [String] -> String
h = foldr (++) [] . filter c

test1c =
h ["Once","Upon","a","Time"] == "OnceUponTime" &&
h ["no","capitals","!" ] == ""     &&
h ["ALL","CAPS"]      == "ALLCAPS"     &&
h ["ab","Cd","Ef","gh","ij"] == "CdEf"

test1 = test1a && test1b && test1c
prop_1 xs = all (not . null) xs ==> f xs == g xs && g xs == h xs
check1 = quickCheck prop_1

-- 2a

p :: [a] -> [a]
p xs = [ x | (i,x) <- zip [0..] xs, i `mod` 3 == 0 ]

test2a =
p ["abcdefgij"] == "adgij" &&
p [1,2,3,4,5] == [1,4] &&
p [0,0,0,0,0] == [0,0] &&
(p [] :: [Int]) == []

-- 2b
\[ q :: [a] \rightarrow [a] \]
\[
q [] = []
\]
\[
q [x] = [x]
\]
\[
q [x,y] = [x]
\]
\[
q (x:y:z:ws) = x : q ws
\]

\[
\text{test2b} = \]
\[
q \ "abcdefghijklmnopqrstuvwxyzij" = \ "adgj" \&
\]
\[
q [1,2,3,4,5] = [1,4] \&
\]
\[
q [0,0,0,0,0] = [0,0] \&
\]
\[
(q [] :: [Int]) = []
\]

\[
\text{test2} = \text{test2a} \&\& \text{test2b}
\]

\[
\text{prop}_2 :: [\text{Int}] \rightarrow \text{Property}
\]
\[
\text{prop}_2 \text{xs} = \text{even} (\text{length} \text{xs}) \implies p \text{xs} = q \text{xs}
\]

\[
\text{check2} = \text{quickCheck} \ \text{prop}_2
\]

-- 3a

\[
data \ \text{Term} = \ \text{Con} \ \text{Int} \]
\[
| \ \ X
\]
\[
| \ \text{Term} :+: \text{Term}
\]
\[
| \ \text{Term} :*: \text{Term}
\]

\[
\text{eva} :: \text{Term} \rightarrow \text{Int} \rightarrow \text{Int}
\]
\[
\text{eva} \ (\text{Con} \ i) \ x = i
\]
\[
\text{eva} \ (X) \ x = x
\]
\[
\text{eva} \ (t :+: u) \ x = \text{eva} \ t \ x + \text{eva} \ u \ x
\]
\[
\text{eva} \ (t :*: u) \ x = \text{eva} \ t \ x \ * \ \text{eva} \ u \ x
\]

\[
\text{test3a} =
\]
\[
\text{eva} \ (\text{Con} \ 3) \ 3 = 3 \&\&
\]
\[
\text{eva} \ (\text{Con} \ 3) \ 5 = 3 \&\&
\]
\[
\text{eva} \ X \ 3 = 3 \&\&
\]
\[
\text{eva} \ X \ 5 = 5 \&\&
\]
\[
\text{eva} \ (X :*: X) \ 3 = 9 \&\&
\]
\[
\text{eva} \ ((X :*: X) :+: \text{Con} \ 1) \ 3 = 10 \&\&
\]
\[
\text{eva} \ ((X :*: (X :+: \text{Con} \ 1))) \ 3 = 12 \&\&
\]
\[
\text{eva} \ ((\text{Con} \ 2 :*: (X :*: X)) :+: ((\text{Con} \ 3 :*: X) :+: \text{Con} \ 4)) \ 5 = 69
\]

-- 3b

\[
\text{sho} :: \text{Term} \rightarrow \text{String}
\]
\[
\text{sho} \ (\text{Con} \ i) = \ \text{show} \ i
\]
\[
\text{sho} \ (X) = "x"
\]
\[
\text{sho} \ (t :+: u) = "(\ ++ \ \text{sho} \ t \ ++ \ +\ ++ \ \text{sho} \ u \ ++ \ )"
\]
\[
\text{sho} \ (t :*: u) = "(\ ++ \ \text{sho} \ t \ ++ \ *\ ++ \ \text{sho} \ u \ ++ \ )"
\]

\[
\text{test3b} =
\]
\[
\text{sho} \ (\text{Con} \ 3) = "3" \&\&
\]
\[
\text{sho} \ (\text{Con} \ 3) = "3" \&\&
\]
\[
\text{sho} \ X = "x" \&\&
\]
\[
\text{sho} \ (X :*: X) = "(\text{x*x})" \&\&
\]
\[
\text{sho} \ ((X :*: X) :+: \text{Con} \ 1) = "((\text{x*x})+1)" \&\&
\]
\[
\text{sho} \ ((X :*: (X :+: \text{Con} \ 1))) = "\text{x*(x+1)}" \&\&
\]
\[
\text{sho} \ ((\text{Con} \ 2 :*: (X :*: X)) :+: ((\text{Con} \ 3 :*: X) :+: \text{Con} \ 4)) = "((2*\text{x*x})+((3*\text{x}+4))"
\]

\[
\text{test3} = \text{test3a} \&\& \text{test3b}
\]
-- all

test = test1 && test2 && test3
check = check1 >> check2