-- Zayadur Khan CIS-252 M001 Assignment 3

series :: Int -> a -> [a]

series n item

| n <= 0 = []

| otherwise = item : series (n-1) item

countUp :: Int -> Int -> [Int]

countUp m n

| m > n = []

| otherwise = m : countUp (m+1) n

-- squarePairs is a recursive function that returns

-- a list of n pairs of the form (x, x^2), where the

-- value of x in the first pair is i, and the

-- value increases by 1 in each subsequent pair

-- return [] if n < 0

squarePairs :: Int -> Integer -> [(Integer, Integer)]

squarePairs n i

| n <= 0 = []

| otherwise = (i, i^2) : squarePairs (n-1) (i+1)

-- countDownBy generates the list of values that starts

-- with m, each subsequent value is obtained by subtracting

-- diff, and the list ends when the next value would be

-- less than n; if m < n, return []

countDownBy :: Int -> Int -> Int -> [Int]

countDownBy m n diff

| m < n = []

| otherwise = m : countDownBy (m-diff) n diff

-- steps returns list containing n-m+1 lists

-- provided m <= n; the ith list is the sequence of values

-- from m to m+1-i; if m > n, return []

-- TODO: prevent printing empty lists; handle m < 0

steps :: Int -> Int -> [[Int]]

steps m n

| m > n = []

| otherwise = helper 1

where

helper i

| i > n = []

| otherwise = countUp m i : helper (i+1)

-- indexChar returns a string of length n, where the ith

-- character is '!', and every other character is c

-- if n < 0, return []

-- if i < 1 || i > n, return list containing instances of c

indexChar :: Int -> Int -> Char -> String

indexChar n i c

| n <= 0 = []

| i < 1 || i > n = series n c

| i == 1 = '!' : indexChar (n-1) (i-1) c

| otherwise = c : indexChar (n-1) (i-1) c

-- diag returns list of n strings of length n, where

-- the ith string contains a '!' in the ith position,

-- and every other character is a copy of c

diag :: Int -> Char -> [String]

diag n c

| n <= 0 = []

| otherwise = helper 1

where

helper i

| i < 1 || i > n = []

| otherwise = indexChar n i c : helper (i+1)a