**CSCE 330 Fall 2016**

Quiz 5

Assigned Thursday 16-11-10

Recall that the Haskell Prelude makes available a function called zip that takes two lists and returns a list of tuples. For example:

ghci> zip ['a'..'z'] [1..]

[('a',1),('b',2),('c',3),('d',4),('e',5),('f',6),('g',7),('h',8),('i',9),('j',10),('k',11),('l',12),('m',13),('n',14),('o',15),('p',16),('q',17),('r',18),('s',19),('t',20),('u',21),('v',22),('w',23),('x',24),('y',25),('z',26)]

This function can be written using pattern matching and recursion. Consider the following implementation of Haskell’s zip function.

zip [] \_ - []

zip \_ [] = []

zip (x:xs) (y:xy) = (x,y):zip xs xy

Write the function type for the zip function.

**Answer:**

zip :: [a] -> [b] -> [(a,b)]