**Group Activity 8: CS 3060**

**Names** of students in your group: Jacob Partin, Riley Pearson

Points: 10

**Task 1**: (3 points) Consider the following Haskell function.

myFunc :: [Int] -> Int

myFunc [] = 1

myFunc (h:t) = 2 \* h \* (myFunc t)

main = print(myFunc [3,1,2])

Guess the output if you call ‘main’. Why so? Write your explanation in a few lines.

When main is called, the output is 48. This Is because myFunc calculates the product of 2 times each element, such that myFunc[3,1,2]=(2\*3)\*(2\*1)\*(2\*2)=48

Then, run the above code and verify whether your guess is correct. Paste a related screenshot below.

A screenshot of a computer

Description automatically generated with medium confidence

**Task 2**: (4 points) Write a Haskell function *foo* which takes a list of characters (Char) and returns the number of characters present in the list. If the list is empty *foo* returns zero.

foo :: [Char] -> Int

foo [] = 0

foo (h:t) = 1 + foo t

**Task 3**: (3 points) Run the following snippet of code, which uses the *list comprehension* technique to build a list.

*players = ["x","y", "z"]*

*matches = [(m, n) | m <- players, n <- players, m < n]*

*main = print(matches)*

Add one line of code to print the content of list *matches*.

Now write a new piece of code to generate all permutations of letters a, b, and c.

letters = ["a","b", "c"]

permutations = [ (j, k, l)

    | j<-letters, k<-letters, l<-letters, j/=k, j/=l, k/=l]

main = print(permutations)

Submission: Submit one copy (per group) on Canvas.