**Group Activity 13: CS 3060**

**Names** of students in your group: Sidney Sanders, Amanda Collert, Quinci Drain, Emily Endlish

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.

2 \* 3 \* 2 \* 1 \* 2 \* 2 \* 1 = 48

Walking through the code, stepping through the recursion to get a equation it equals to 48.

Then, run the above code and verify whether your guess is correct.

48

**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)

main :: IO ()

main= do

print(foo ['a', 'b', 'c', 'd'])

print(foo [])

> .\task1.exe

4

0

**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]*

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

main = do print(matches)

> .\task1.exe

[("x","y"),("x","z"),("y","z")]

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

perm = [(m, n) | m <- letters, n <- letters, m /= n]

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

main = do print(perm)

> .\task1.exe

[("a","b"),("a","c"),("b","a"),("b","c"),("c","a"),("c","b")]

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