Université d'Ottawa
Faculté de génie

École de science
d'informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical
Engineering
and Computer Science

Assignment 3
CSI2120 Programming Paradigms
Winter 2020
Due on April 5th before 11:00 pm in Virtual Campus
6 marks

There are [11 points] in this assignment. The assignment is worth 6% of your final mark.

All code must be submitted in a scm file. Screenshots, files in a format of a word editor, pdfs, handwritten solutions, etc. will not be marked and receive an automatic 0.

Question 1. Filter and Map **[1 point]**

Use the built-in filter and map to change a list as follows
Drop each number between -1 and +1 inclusive
For each number greater than 1 replace the number with 10 times the number.
For each number smaller than -1 replace the number with the absolute value of the reciprocal.

```
(changeList '(0 -2 3 -4 1))
⇨ '( 1/2 30 1/4 )
```

Question 2. List Processing **[2 points]**

Find the longest sub-list of numbers which are identical and return this sub-list.
In case of a tie, return the sub-list occurring last.

```
(sameNum '( 0 1 5 3 3 3 2 1 1))
⇨ '( 3 3 3)

(sameNum '( 0 1 5 3 3 3 2 1 1 1))
⇨ '( 1 1 1)
```

Question 3. Travellers **[4 points]**

Implement a function `destination` that tallies the vote for where a group of friends want to travel to. Each friend votes for three destinations. The result needs to be a list containing all destinations that received the most votes (ties are possible).

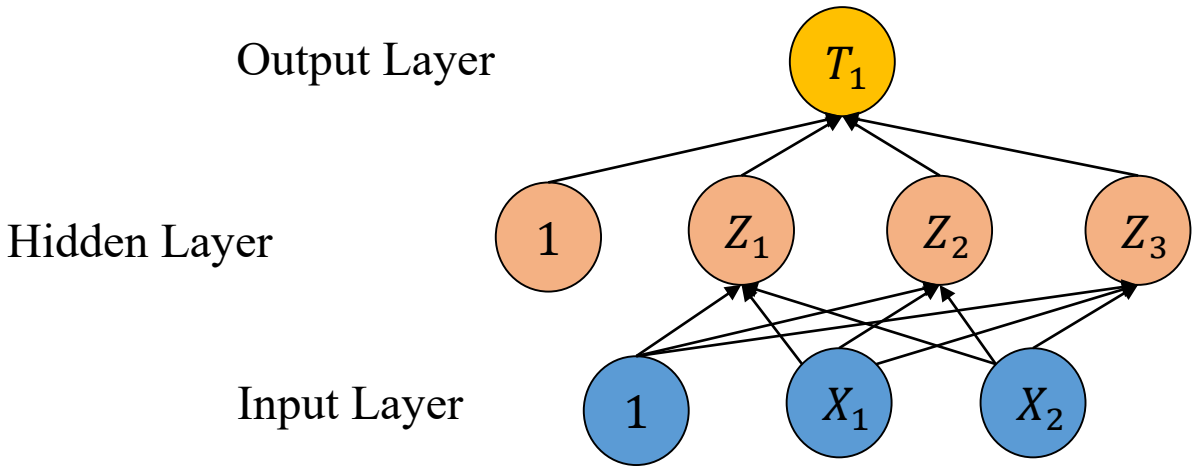
Examples:

```
(define choices '('("marie" '("peru" "greece" "vietnam"))
                  '("jean" '("greece" "peru" "vietnam"))
                  '("sasha" '("vietnam" "peru" "greece"))
                  '("helena" '("peru" "vietnam" "greece"))
                  '("emma" '("greece" "peru" "vietnam"))))

(destination choices)
⇨ '(("peru" . 13))


(define choices '('("marie" '("peru" "greece" "vietnam"))
                  '("jean" '("greece" "peru" "vietnam"))
                  '("sasha" '("vietnam" "peru" "greece"))
                  '("helena" '("peru" "vietnam" "greece"))
                  '("emma" ("greece" "peru" "vietnam"))
                  '("jane" ("greece" "vietnam" "peru"))))

(destination choices)
⇨ '(("peru" . 13) ("greece" . 13))
```

You are allowed extra global helper functions.

Question 4. Neural Network **[4 points]**

Write a program that simulates a Neural Network in Scheme. We use what is called a multilayer perceptron with 1 hidden layer with an input and output layer. Each neuron calculates the weighted sum of its input (including an offset) and applies an activation function. Consider the figure below. for the example of a slice of size 2.



In the above example the equation for the first hidden neuron $Z_1 = \sigma(\alpha_{10} + \alpha_{11}X_1 + \alpha_{12}X_2)$. The equation for the output neuron is $T_1 = \sigma(\beta_{10} + \beta_{11}Z_1 + \beta_{12}Z_2 + \beta_{13}Z_3)$**.** The inputs X_1 and X_2 simply pass their value to all hidden layers.

Your program must create exactly he network shown above with the following parameters

$$\alpha_{10} = 0.1 \;\; \alpha_{11} = 0.3 \;\; \alpha_{12} = 0.4$$
$$\alpha_{20} = 0.5 \;\; \alpha_{21} = 0.8 \;\; \alpha_{22} = 0.3$$
$$\alpha_{30} = 0.7 \;\; \alpha_{31} = 0.6 \;\; \alpha_{32} = 0.6$$

$$\beta_{10} = 0.5 \;\; \beta_{11} = 0.3 \;\; \beta_{12} = 0.7 \;\; \beta_{13} = 0.1$$

a) Implement a neural network node which takes a list of weights and an activation function. The function is to return a function accepting a list of input values.

```
(neuralNode '(0.1 0.3 0.4) sigmoid)
⇨ #<procedure>

((neuralNode '(0.1 0.3 0.4) sigmoid) '(0.5 0.5))
⇨ 0.610639233949222
```

b) Use your `neuralNode` function to implement one layer of the neural network shown in the above figure. Use the sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$ as the activation function.

Example (hidden layer)

```
(neuralLayer '((0.1 0.3 0.4)(0.5 0.8 0.3)(0.7 0.6 0.6)))
⇨ #<procedure>

((neuralLayer '((0.1 0.3 0.4)(0.5 0.8 0.3)(0.7 0.6 0.6))) '(0.5
0.5))
⇨ '(0.610639233949222 0.740774899182154 0.7858349830425586)
```

c) Use your `neuralLayer` function to implement the complete neural network shown in the above figure. Use the sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$ as the activation function.

```
(neuralNet '(0.5 0.5))
⇨ '(0.782503850784443)
```

d) Write a function that takes as input a parameter k determining the number of times the neural network is run. Your program must apply the neural network to a series of input values and calculate the network output for each pair of input values in a loop. These input values are $X_{1,k} = \sin\frac{2\pi(k-1)}{N}$ and $X_{2,k} = \cos\frac{2\pi(k-1)}{N}$ where $k = 1 \ldots N$ is the index of the current loop variable. The function must return all calculated output values in a list.

```
(applyNet 16)
⇨ '(0.7770060393971061 0.7853501263409901 0.7897263070205873
   0.7902968594617558 0.7872242379440337 0.7805159297515977
   0.7702872231675277 0.7573421434408293 0.7436542255881534
   0.7320120209466757 0.7248686717891844 0.7235344799330423
   0.7281858924381838 0.7380276807952172 0.7512282131381334
   0.765101793865149)
```

Note that your output formatting may look different (number of digits and or line breaks)