Language and Speech Technology - Assignment 1

Gonçalo Carvalho (s3450295) & Timen van Gelderen (s3427781)

September 9, 2019

1 waves.py

```
#!/usr/bin/python3
#waves.py
#Goncalo Carvalho (s3450295)
#Timen van Gelderen (s3427781)

# add necessary imports
import sys
from numpy import *
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import math
import pylab

def periodic(t, f, A, phi):
    # calculate output
    # Calculate timestamps

    # a. ------------------------------------------------------
    # use linspace to get the timestamps (returns an array)
    # grab the ones that have an fs of 44100
    c = 0
    for i in t:
        c+= 1

    x = np.zeros((c))
    c=0
    for i in t:
        theta = 2 * math.pi * float(f) * i + phi
        x[c] = (round(float(A) * math.cos(theta),8))
        c+=1
    return x
```

```
# This function creates a plot with two lines, representing two
# periodic functions with ‘Time (s)’ on the x-axis and ‘Amplitude’
# on the y-axis.
def plot_wave(t1, x1, t2, x2):
    # First function
    plt.plot(t1, x1, label="x1(t)", color='r')
    plt.plot(t2, x2, label="x2(t)", color='blue')
    plt.axhline(y=0, color='black', linestyle='-')
    plt.title("Phase comparision")
    plt.xlabel("Time (s)")
    plt.ylabel("Amplitude (A)")
    plt.show()

if __name__ == '__main__':
    dur = sys.argv[1]
    # to read values as long as there is input
    # calculate t, and extract the values for f, A, and phi
    t = np.linspace(0, float(dur), num=44100)
    # b. ------------------------------------------------------
    f =  sys.argv[2]
    # checking if A and phi are null and changing them accordingly

    if len(sys.argv) > 4:
        A = sys.argv[3]
        phi = float(sys.argv[4]) * math.pi
    else:
        if len(sys.argv) > 3:
            A = sys.argv[3]
        else:
            A = 1
        phi = 0


    # ------------------------------------------------------------
    x = periodic(t, f, A, phi)
    # do something with x
    x1 = periodic(t, f, A=1.3, phi=0)
    x2 = periodic(t, f, A=1.2, phi=0.9)
    x3 = x1 + x2
    plot_wave(t, x3, t, x)
```

2 sum.py

```
#!/usr/bin/python3
# sum.py
```

```
#Goncalo Carvalho (s3450295)
#Timen van Gelderen (s3427781)

# add necessary imports
import sys
from numpy import *
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import math
import pylab

def periodic(t, f, A, phi):
    # calculate output
    # Calculate timestamps

    # a. ------------------------------------------------------
    # use linspace to get the timestamps (returns an array)
    # grab the ones that have an fs of 44100
    c = 0
    for i in t:
        c+= 1

    x = np.zeros((c))
    c=0
    for i in t:
        theta = 2 * math.pi * float(f) * i + phi
        x[c] = (round(float(A) * math.cos(theta),8))
        c+=1
    return x

def sum_waves(A_array, Phi_array):
    rcos=0
    rsin=0
    i=0
    for A in A_array:
        rcos += A*cos(Phi_array[i])
        rsin += A*sin(Phi_array[i])
        i+=1
    a = rcos*rcos + rsin*rsin
    phi = math.atan(rsin/rcos)

    return a, phi

# This function creates a plot with two lines, representing two
# periodic functions with ‘Time (s)’ on the x-axis and ‘Amplitude’
```

```
# on the y-axis.
def plot_wave(t1, x1, t2, x2):
    # First function
    plt.plot(t1, x1, 'r')
    plt.plot(t2, x2, 'blue')
    plt.axhline(y=0, color='black', linestyle='-')
    plt.title("Phase comparision")
    plt.xlabel("Time (s)")
    plt.ylabel("Amplitude (A)")
    plt.legend()
    plt.show()

if __name__ == '__main__':
    arg1 = sys.argv[1].split(',')
    amplitudes = [float(x) for x in arg1]

    arg2 = sys.argv[2].split(',')
    phases = [float(x) for x in arg2]

    print("input:\nA=", amplitudes, ", Phi=", phases)

    a, p = sum_waves(amplitudes, phases)
    print("output:\nA=", a, ", Phase=", p)
```