Project 4 – Word Puzzle

Name: Claire Minahan

Instructor: S. Einakian

Section: CPE101-04

funcs.py

string → list(list)

def make_rows(string1):

takes intervals of 10 from the string and inputs them into individual lists, representing each row

puts all of the individual lists into one large list named ‘rows’

string→ list(list)

def make_columns(string1):

takes every 10 letters (with different start positions) from the string and inputs them into individual lists, representing each column

puts all the individual lists in to one large list named ‘columns’

list string → boolean string

def find_horizontal_forward(rows, word):

goes through a list of rows and searches for a word

checks if the current character is the same as the first character of the word you’re searching for

if so, goes onto next item in list and compares to next character in word

if number of matches is same as length of word, returns location of word

otherwise informs that the word is not in a row going forward

list string → boolean string

def find_horizontal_backwards(rows, word):

goes through a list of rows and searches for a word

goes backwards through the current list in rows and checks if the current character is the same as the first character of the word you're searching for

if so, goes onto next item in list and compares to the next character in word

if number of matches is same as length of word, returns location of word

otherwise informs that the word is not in a row going backwards

list string → boolean string

def find_vertical_down(cols, word):

goes through a list of columns and searches for a word

checks if the current character is the same as the first character of the word you're searching for

if so, goes onto next item in list and compares to next character in word

if number of matches is same as length of word, returns location of word

otherwise informs that the word is not in a column going down

list string → boolean string

def find_vertical_up(cols, word):

goes through a list of rows and searches for a word

goes backwards through the current list in column and checks if the current character is the same as the first character of the word you're searching for

if so, goes onto next item in list and compares to the next character in word

if number of matches is same as length of word, returns location of word

otherwise informs that the word is not in a column going down