```
'''
    ACSL Difference Factor Problem:
    Given two strings, calculates the ACSL difference factor (ADF)

    Author: Rumaisa Abdulhai
'''

#####################
# IMPORT STATEMENTS #
#####################
import numpy as np

####################
# GLOBAL VARIABLES #
####################
adf = 0 # ACSL DIFFERENCE FACTOR

####################
# PROCESSING LINES #
####################
def pre_process(file):
    '''
    Takes in a text file and groups sentence pairs in a list.
    Ignores all non-alphabetic characters and converts all letters to uppercase.

    Parameters:
    ----------
    file: The text file
    '''

    ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

    lines = [line.upper() for line in file]
    groups = [ [lines[i],lines[i-1]] for i in range(1,len(lines),2) ]

    sentence_pairs = []
    for group in groups:
        pairs = []
        for sentence in group:
            for character in sentence:
                if character not in ALPHABET:
                    sentence = sentence.replace(character,'')

            pairs.append(sentence)
        sentence_pairs.append(pairs)

    return sentence_pairs

######################
# FIND COMMON STRING #
######################
def get_longest_string(x,y):
    '''
    Takes in two strings. Returns the longest common
    substring contained in each string.

    Parameters:
    ----------
    x: The first string\n
    y: The second string
    '''

    global adf
    first, second = x, y

    num_rows = len(first) + 1
```

```
    num_cols = len(second) + 1
    common_strings = np.zeros((num_rows, num_cols), dtype=int).tolist()

    longest_len, index = 0, 0
    for i in range(1, num_rows):
        for j in range(1, num_cols):
            if first[i - 1] == second[j - 1]:
                common_strings[i][j] = common_strings[i - 1][j - 1] + 1
                if common_strings[i][j] > longest_len:
                    longest_len = common_strings[i][j]
                    index = i
            else:
                common_strings[i][j] = 0

    common = first[index - longest_len: index]

    max = 0
    for i in range(len(common_strings)):
        for j in range(len(common_strings[0])):
            if common_strings[i][j] > max:
                max = common_strings[i][j]

    if max != 0:
        best_common = common
        for i in range(len(common_strings)):
            for j in range(len(common_strings[0])):
                if common_strings[i][j] == max:
                    first_index = i - max
                    first_string = first[first_index : first_index + max]

                    if first_string < best_common:
                        best_common = first_string

        common = best_common

    if len(common) != 0:

        # print("COMMON:", common, "      LENGTH OF COMMON:",len(common))
        adf += len(common)

        left_first = first.replace(common,".", 1).split(".")[0]
        # print("left first:", left_first)

        left_second = second.replace(common,".", 1).split(".")[0]
        # print("left second:", left_second)

        right_first = first.replace(common,".", 1).split(".")[1]
        # print("right first:", right_first)

        right_second = second.replace(common,".", 1).split(".")[1]
        # print("right second:", right_second)

        get_longest_string(left_first,left_second)
        get_longest_string(right_first,right_second)

##########################
# ACSL DIFFERENCE FACTOR #
##########################
def acsl_difference_factor(file_path):
    '''
    Given a file of strings, prints the ACSL
    difference factor (ADF) for each set of strings.

    Parameters:
    ----------
    file_path: The file path as a String.
    '''
```

```
134
135     global adf
136     textFile = open(file_path)
137     sentence_pairs = pre_process(textFile)
138
139     for groups in sentence_pairs:
140         get_longest_string(groups[1],groups[0])
141         print(adf)
142         adf = 0
143
144 ################
145 # MAIN PROGRAM #
146 ################
147 def main():
148     file_path = 'ACSL/sampleinput.txt'
149     acsl_difference_factor(file_path)
150
151 ####################
152 # START OF PROGRAM #
153 ####################
154 if __name__ == '__main__':
155     main()
```