# **Searching Strings**

# **Model Answers**

**Challenge 1**

Write a program that returns the number of spaces in the string.

# SearchingStringsChallenge1.py

# Challenge 1

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

print("The number of times that a space appears is {0}\n"

.format(fight\_club.count(" ")))

'''

#assertions

String: fight\_club = "The first rule of fight club is: do not talk about fight club."

Output: The number of times that a space appears is 12

'''

**Challenge 2**

Write a program that returns the total number of punctuations (full stops and colons) used

# SearchingStringsChallenge2.py

# Challenge 2

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

print("The number of times that punctuation appears is ", (fight\_club.count(",") + fight\_club.count(".") + fight\_club.count(":")))

'''

#assertions

String: fight\_club = "The first rule of fight club is: do not talk about fight club."

Output: The number of times that a space appears is 2

'''

**Challenge 3**

Write a program that gets user input and determines whether the text matches the start of the string.

# SearchingStringsChallenge3.py

# Challenge 3

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

user\_text = input("Please enter some case sensitive text.\n\n")

print("Does this string start with the given text?....{0}\n"

.format(fight\_club.startswith(user\_text)))

'''

#Assertions

Input: the

Output: Does this string start with the given text?....False

#assertion 2

Input: The

Output: Does this string start with the given text?....True

'''

**Challenge 4**

Write a program that gets user input and determines whether the text matches the start or the end of the string.

# SearchingStringsChallenge4.py

# Challenge 4

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

user\_text = input("Please enter some case sensitive text.\n\n")

print("Does this string start with the given text?....{0}\n"

.format(fight\_club.startswith(user\_text)))

print("Does this string end with the given text?....{0}\n"

.format(fight\_club.endswith(user\_text)))

'''

#Assertions

Input: The f

Output:

Does this string start with the given text?....True

Does this string end with the given text?....False

#assertion 2

Input: club.

Output:

Does this string start with the given text?....False

Does this string end with the given text?....True

'''

**Challenge 5**

Write a program that returns the index position of the first instance of the text "fight".

# SearchingStringsChallenge5.py

# Challenge 5

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

fight\_club\_fight\_index = fight\_club.find("fight")

print("The index position of fight is {0}\n"

.format(fight\_club\_fight\_index))

'''

Input: fight\_club = "The first rule of fight club is: do not talk about fight club."

Output:

The index position of fight is 18

'''

**Challenge 6**

Write a program that returns the position of the instance of the text "fight" that appears after the colon.

# SearchingStringsChallenge6.py

# Challenge 6

# Author: A. N. Other

# date: November 2016

fight\_club = "The first rule of fight club is: do not talk about fight club."

fight\_club\_colon\_index = fight\_club.find(":")

fight\_position = fight\_club.find("fight", fight\_club\_colon\_index, len(fight\_club))

print("Looking for the word fight, anytime after :...\n\n"

"The word fight appears at index position..{0}"

.format(fight\_position))

'''

#assertion

Input: fight\_club = "The first rule of fight club is: do not talk about fight club."

Output:

Looking for the word fight, anytime after :...

The word fight appears at index position..51

'''

**Challenge 7**

Write a program that gets a username from a stored email address. The email address should be a person's first name, followed by a dot, followed by their last name. The username should be the person's last name followed by their first initial. Here is an example test case assertion:

"robert.paulson@hotmail.com" produces the username "paulsonr"

# SearchingStringsChallenge7.py

# Challenge 7

# Author: A. N. Other

# date: November 2016

email\_address = "robert.paulson@hotmail.com"

dot\_index = email\_address.find(".")

at\_index = email\_address.find("@")

username\_surname = email\_address[dot\_index+1:at\_index]

username = username\_surname + email\_address[0]

print(username)

'''

#assertion

Input: email\_address = "robert.paulson@hotmail.com"

Output: paulsonr

'''

**Challenge 8**

Write a program that contains a stored pin as a string. The program must evaluate the string and say whether it contains numbers only.

# SearchingStringsChallenge8.py

# Challenge 8

# Author: A. N. Other

# date: November 2016

pin = "2580"

if pin.isdigit() == True:

print("This pin is made up only of digits\n\n")

else:

print("This pin contains characters other than digits\n\n")

'''

#assertions

Input: pin = "25.0"

Output: This pin contains characters other than digits

Input: pin = "25r0"

Output: This pin contains characters other than digits

Input: pin = "2580"

Output: This pin is made up only of digits

'''

**Challenge 9**

Write a program that gets a password string from the user. The program must evaluate the string to ensure that it contains a mixture of numbers, letters, and uppercase letters. The password must be no less than 8 characters.

# SearchingStringsChallenge9.py

# Challenge 9

# Author: A. N. Other

# date: November 2016

password = input("Please enter a password:\n\n")

contains\_digits = False

contains\_upper = False

if len(password) < 8:

print("The password needs to contain at least 8 characters")

if any(c.isalpha() for c in password) == False:

print("The password needs to contain at least one alphabetical character")

if any(c.isdigit() for c in password) == False:

print("The password needs to contain at least one digit")

if any(c.isupper() for c in password) == False:

print("The password needs to contain at least one uppercase character")

'''

#assertions

Input: hello

Output:

The password needs to contain at least 8 characters

The password needs to contain at least one digit

The password needs to contain at least one uppercase character

Input: hellohello

Output:

The password needs to contain at least one digit

The password needs to contain at least one uppercase character

Input: hello123

Output:

The password needs to contain at least one uppercase character

'''

**Challenge 10**

Write a program that takes a list of email addresses and iterates it to produce a list containing lists of first names and last names using title case. Assume that all email addresses use the format: firstname.surame@somewhere.something

Here is a test case assertion:

list\_1 = ["tyler.durden@gmail.com", "marla.singer@yahoo.co.nz"] should produce

list\_2 = [['Tyler', 'Durden'], ['Marla', 'Singer']]

# SearchingStringsChallenge10.py

# Challenge 10

# Author: A. N. Other

# date: November 2016

list\_1 = ["tyler.durden@gmail.com", "marla.singer@yahoo.co.nz"]

list\_2 = []

for count in range (0, len(list\_1)):

dot\_index = list\_1[count].find(".")

at\_index = list\_1[count].find("@")

first\_name = list\_1[count][0:dot\_index]

last\_name = list\_1[count][dot\_index + 1:at\_index]

title\_first\_name = first\_name.title()

title\_last\_name = last\_name.title()

list\_3 = []

list\_3.append(title\_first\_name)

list\_3.append(title\_last\_name)

list\_2.append(list\_3)

print(list\_2)

'''

#assertion

Input:

list\_1 = ["tyler.durden@gmail.com", "marla.singer@yahoo.co.nz"] should produce

Output:

list\_2 = [['Tyler', 'Durden'], ['Marla','Singer']]

'''