# Worksheet 4 Regular expressions Answers

1. **A simple Regular Expression**

**Extension**: Try adding a loop so that the user will keep being asked to re-enter their name until they do it right

import re

name = input("Enter your name: ")

valid = re.match("[A-Z]", name)

while not valid:

print("Invalid, no capital")

name = input("Please re-enter your name: ")

valid = re.match("[A-Z]", name)

print ("That looks OK")

***See Program L4 WS4 Ex1 name number validation.py in Sample Programs folder***

1. **Checking for lower case letters or digits**

(b) Now write a program that asks for an input that starts with a number (perhaps a phone number).

import re  
 number = input(“Enter your phone number: ”)  
 valid = re.match(“[0-9]”,number)  
 if valid:  
 print(“That looks OK”)  
 else:  
 print(“Erm, try again!”)

***See Program L4 WS4 Ex2 phone number validation.py in Sample Programs folder***

1. **A complex Regular Expression**

(a) Alter the program so that it accepts postcodes with one or two letters and one or two numbers at the start, followed by a space and one number followed by two letters. Test your programs with each of these postcodes:

import re  
 code = input(“Enter your postcode: ”)  
 valid = re.match(“[A-Z]{1,2}[0-9]{1,2} [0-9][A-Z][A-Z]”,code)

# or, insert \s instead of leaving a space in the middle,

# and could use d instead of [0-9]  
 if valid:  
 print(“That looks OK”)  
 else:  
 print(“Erm, try again!”)

TS16 1DA - should be VALID (two letters and two numbers at the start)  
DT2 7EW - should be VALID (two letters and one number at the start)  
W1 8BL - should be VALID (one letter and one number at the start)  
C17 9DF - should be VALID (one letter and two numbers at the start)  
E 9GL - should be INVALID (no number in the first part)  
N32 178 - should be INVALID (no letters in the second part)

EC1A 1CD - will be judged invalid, although this is a valid postcode.

The validation routine still needs some work!

***See Program L4 WS4 Ex3a postcode validation.py in Sample Programs folder***

(b) **Extension:** Complete the following program to allow the user to test as many postcodes as they like, stopping when they press **Enter** instead of a postcode.

**Note:** The validation rule is simplified in this program, it just checks that the postcode starts with 2 uppercase letters followed by a number.

#Worksheet 1 Qu 3: Program tests postcodes according to the rule,

#"starts with 2 capital letters then a number"

import re

print ("Test as many postcodes as you like.. press Enter to end.")

anotherGo = True

while anotherGo:

postcode = input("Please input your postcode: ")

valid = re.match("[A-Z][A-Z][0-9]",code) #INSERT LINE HERE

if len(postcode)==0:

anotherGo = False #INSERT LINE HERE

else:

if valid: #INSERT CONDITION HERE

print("valid postcode")

else:

print("invalid postcode")

***See Program L4 WS4 Ex3b postcode validation.py in Sample Programs folder***

**4. Extension: A regular expression to validate any postcode**

To make it work for all postcodes:

valid = re.match("[A-Z]{1,2}\d{1,2}[A-Z]?\s\d[A-Z][A-Z]",postcode)

or a variation on this, e.g. [A-Z]{1,2}[0-9]{1,2}[A-Z]? [0-9][A-Z][A-Z]”,

Another way of validating a postcode would be to use 6 different patterns and an IF..ELIF construct.

***See Program L4 WS4 Ex4 postcode validation.py in Sample Programs folder***