The 4 characteristics of a strong password :

one lowercase letter; ONE UPPERCASE; 1 digit; morethan7characters

Abstraction

We want to develop an algorithm that will solve the problem initially for just one password from the user.

Some of the key components of the problem are :

- The password will be represented by a string
- Each character in the string will need to be examined to determine if it is uppercase, lowercase or a digit

Writing your Thinking

Take 5 minutes to think about how you would tackle this problem.

- **Would you check the length first so you would not have go any further? What does this mean for assessing the overall strength?**
- If a character is lower case, how will you store that information? What will you call the variable?
- **How will you accumulate the strength of the string?**
- What are good examples to enter as the user to test your program? Would it be better if your partner tested your program and you tested your partner's?

Using Think-Pair-Share-Square (TPSS), go through how you and your partner were thinking about how to solve the problem.

Pseudo-Code

#Initialise the conditions

Set all 4 characteristics to False;

Ask the user for their password;

#Check each character of the string if it meets a strong characteristic

If length of password >7 { long_enough_password = True}

for i = 1 to length of password {

    if password(i) == uppercase { oneUpperCaseChar = True }

    else if password(i) == lowercase { one_lower_case_char = True }

    else if password(i) == digit { one_digit_char = True }

}

Output the strength of the password from 0-4.

An extension of the challenge

There are 4 key characteristics of a strong password.

Can you tell the user which characteristics are present and which characteristics are missing from their password?

For example, Your password contains more than 8 letters, at least one lower case letter and digit, but there are no uppercase letters.