Workshop

Week 40

Warm-up

Create a function named countVowels that counts the number of vowels in an English word.

The function should take one input, an English word, and should return the number of vowels in that word.

Remember proper function documentation!

Exercise 1a

Use the vowel counting program in exercise 1a from last week's workshop and convert the program into a program run only by functions.

The program should consist of three functions:

- countVowels – return the number of vowels in a given word.
- getWord – return the word prompted from an user.
- main – calls all the functions and displays the result to the user.

All functions should be properly documented, and the program should *call* the main function in order to execute the program.

Exercise 1b

Use the vowel counting program in exercise 1c from last week's workshop and convert the program into a program run only by functions.

The program should create a function that counts the vowels in a word (or import the existing function from the warm-up exercise) and should

the countVowels program from the warm-up exercise.

The program should consist of two functions:

- countVowels – return the number of vowels in a given word (you can import existing function from the warm-up exercise).
- main – prompt the user for words and display the number of vowels until the user decides to quit.

All functions should be properly documented, and the program should make a function call to the main function in order to execute the program.

Exercise 2a

Create a function called randomCharacter that takes a string and that returns a random character from that string.

For instance, if the function is given the string ‘abcdefghijklmnopqrstuvwxyz’, the function should return a random letter from the alphabet.

Remember proper function documentation.

1. Write an algorithm for the function.

2. Implement the function in Python (Hint: use the randint function from the random module).

Exercise 2a

Solution proposal for algorithm:

Input: A sequence of characters of length n with index from 0 to n-1
Output: A random character from the sequence

```
r ← random number between 0 and n-1
Return the character at position r in the sequence
```

Exercise 2b

Write a program that generates a random password of a *specified length*. The password should consist of random lower-case letters, and it should end with a random digit (0-9) and a random symbol (+-*/?!@#$%&).

The program should be run only by functions. Create a function called makePassword that generates a random password of a given length, and generate a random password of length 8 using a main function.

Remember proper function documentation.

1. Write an algorithm for the program.
2. Implement the program in Python.

Exercise 2b

Solution proposal for algorithm:

Input: password_length
Output: random password of requested length

```
password ← random string of letters of length password_length – 2
Randomly draw a digit (0-9) and insert at the end of password
Randomly draw a symbol (+-*/?!@#$%&) and insert at the end of password
Return password
```

Exercise 2c

Modify the password-generating program in the previous exercise so that the symbol and digit are instead inserted at *random* locations in the password except for the first location (the passwrod cannot start with a symbol or a digit).

Implement this change by adding another function called insertRandom to the program.

1. Write an algorithm for the new function.
2. Implement the function in Python.

Exercise 2c

Solution proposal for algorithm:

Input: a string to be modfied (string) and a string to insert (to_insert) that are both zero-based indexed

Output: the original string with the new string inserted at random location (new_string)

```
n ← length of string
r ← a random number between 1 and n
substring_start ← characters in string from location 0 to r
if r < n then
    substring_end ← characters in string from location r to n
else then
    substring_end ←an empty string
end if
new_string ← concatenate substring_start, to_insert and substring_end
Return new_string
```