Exercise 3 – Flow control

Objective

To use the flow control structures of Python and to gain familiarity in coding based on indentation. It takes a little practice! We'll be using a couple of modules from the Python standard library.

Questions

1. Write a Python program that emulates the high-street bank mechanism for checking a PIN. Keep taking input from the keyboard (see below) until it is identical to a password number which is hard-coded by you in the program.

   To output a prompt and read from the keyboard:

   supplied_pin = input("Enter your PIN: ")

   Restrict the number of attempts to three (be sure to use a variable for that, we may wish to change it later), and output a suitable message for success and failure. Be sure to include the number of attempts in the message.

Optional extension

Passwords, and PINs, would not normally be displayed (*echoed*) to the screen for security reasons. So, now we will add the functionality to hide the characters typed. That could be a lot of work, but one of the advantages of using a language like Python is that 'there's a module for it'.

You'll need to **import** a module called **getpass**, which is part of the standard library.

Instead of **input** use **getpass.getpass**, in the same place in the program, with the same parameters.

Note: You will have to run your program in Pycharm or VSCode.

2. Write a Python program to display a range of numbers by steps of -2.
    a) Prompt the user at the keyboard for a positive integer using:

       var = input ("Please enter an integer: ")

    b) Validate the input **(var)** to make sure that the user entered an integer using the **isdecimal()** method. If the user entered an invalid value, output a suitable error message and exit the program.

    c) Use a loop to count down from this integer in steps of 2, displaying each number on the screen until either 1 or 0 is reached. For example, if the integer 16 (validated) is entered, the output would be:

       16
       14
       12
       10
       8
       6
       4
       2
       0

       And if 7 is entered, the output would be:

       7
       5
       3
       1

       You will need to look-up the **range()** built-in in the online documentation, pay particular attention to the *stop* parameter.

If time allows...

3. If a year is exactly divisible by 4 but not by 100, the year is a leap year. There is an exception to this rule. Years exactly divisible by 400 are leap years. The year 2000 is a good example.

   Write a program that asks the user for a year and reports either a leap year or *not* a leap year. (*Hint*: x % y is zero if x is exactly divisible by y.) Test with the following data:

   1984 is a leap year
   1981 is NOT a leap year
   1904 is a leap year
   1900 is NOT a leap year
   2000 is a leap year
   2010 is NOT a leap year

   Use the following to ask the user for a year:

   ```
   year = int(input('Please enter a year: '))
   ```

Solutions

Question 1

There are several valid ways to write this code. Here's one solution:

```
import sys

PIN = '0138'
LIMIT = 4

for tries in range(1, LIMIT):
   supplied_pin = input('Enter your PIN: ')
   if supplied_pin == PIN:
      print('Well done, you remembered it!')
      print('... and after only', tries, 'attempts')
      break
# Note the else: is indented with the for loop, not the if!
else:
   print('You had', tries, 'tries and failed!')
```

Note that we used **uppercase** as a convention for constants, and we took advantage of the **else** on a **for** loop that is *not* executed on a **break**.

Optional extension to Question 1

Using **getpass**, which is part of the standard library:

```
import sys
import getpass

PIN = '0138'
LIMIT = 4

for tries in range(1, LIMIT):
    supplied_pin = getpass.getpass('Enter your PIN: ')
    if supplied_pin == PIN:
        print('Well done, you remembered it!')
        print('... and after only' , tries, 'attempts')
        break
# Note the else: is indented with the for loop, not the if!
else:
    print('You had', tries, 'tries and failed!')
```

Why didn't we use **getpass** in the main question?

Because making the input invisible makes debugging more difficult.

Question 2

Here's one simple solution using the range function:

```
var = input("Please enter an integer: ")

if not var.isdecimal():
    print("Invalid integer:", var)
    exit(1)

for var in range(int(var), -1, -2):
    print(var)
```

Question 3

Here's our solution to test for leap years:

```
y = int(input('Please enter a year: '))

if y%4 == 0 and (y%400 == 0 or y%100 != 0):
    print("Leap Year")
else:
    print("NOT a leap year")
```