CSC1015F Assignment 0: Introduction

Assignment Instructions

The first two questions in this assignment give you practice in typing in and editing programs. The next two are slightly more substantial. Question three involves fixing errors in a given program. Question for concerns a program that consists of the right statements but in the wrong order.

Question 1 [20 marks]

Copy the following program into a file called `secret.py` and test that it runs. The program must be identical, down to the last bit of punctuation, including the spaces at the beginning of some lines!

You may need to create directories as per the instructions in the orientation manual.

```
# program to guess a secret number
# hussein suleman
# 10 february 2011

secret_number = 42    # create secret number in program
guess = 0             # variable to store user's guess

# as long as we have not found the secret number
while guess != secret_number:
    # get a new guess from user
    guess = eval(input("? "))
    # check if guess is too low
    if guess < secret_number:
        print ("lo")
    # or too high
    elif guess > secret_number:
        print ("hi")

print ("Correct!")    # print message indicating success
```

This program is a classic from the early days of Computer Science. A user is expected to guess numbers until he or she converges to a secret internal number. At each incorrect guess, the system lets the user know if the number is too high or too low.

Sample Input/Output:

```
? 22
lo
? 55
hi
? 42
Correct!
```

Question 2 [20 marks]

Edit the program from Question 1 so that the messages printed are more user-friendly. You need to copy the secret.py file from the first question to a file called `secret_2.py`.

Change each of the messages printed to the screen to be the same as the example output below.

User-friendliness of programs was a concept that gained popularity in the 1980s, where programs were made easier for human beings to identify with. This has since grown into the current field of Usability Engineering, which you will learn about while studying Computer Science.

Sample Input/Output:

```
What is the secret number? 14
That is way too low. Please try again.
What is the secret number? 337
That is much too high. Please try again.
What is the secret number? 48
That is much too high. Please try again.
What is the secret number? 40
That is way too low. Please try again.
What is the secret number? 0
That is way too low. Please try again.
What is the secret number? 42
Congratulations, you have guessed the secret number!
```

Question 3 [30 marks]

You will find the following program on the Vula page for this assignment. It is called 'temperature.py'.

```
# This program accepts a temperature given in Fahrenheit and
# prints the equivalent in Celsius and in Kelvin.
# Name: Stephan Jamieson
# 27th February 2017
#
#input_str = input("Enter a temperature in Fahrenheit: ")
fahrenheit_value = eval(input_str)

Calculate equivalent Celsius value
(fahrenheit_value - 32)*(5/9) = celsius_value

# Calculate equivalent Kelvin value
kelvin_value = Celsius_Value+273.15

print(celsius_value, "The temperature in Celsius is:")
print("The temperature in Kelvin is: kelvin_value")
```

The program contains 5 mistakes. They concern the form of comments versus program statements (2 mistakes), the form of assignment statements (1 mistake), the form of print statements (2 mistakes), and the consistent spelling of identifiers (1 mistake).

Here is an example of how the program is meant to behave:

```
Enter a temperature in Fahrenheit: 932
The temperature in Celsius is: 500.0
The temperature in Kelvin is: 773.15
```

Download the program and fix the errors.

NOTE: The following web page contains temperature conversion data you can use to test your solution:

https://en.wikipedia.org/wiki/Conversion_of_units_of_temperature

Question 4 [30 marks]

You will find the following program on the Vula page for this assignment. It is called 'time.py'.

```
# Program to convert an amount of minutes into an equivalent amount
# of days, hours and minutes.
#
# Name: Stephan Jamieson
#
minutes = int(input_str)
days = hours//24

print(".")

hours = minutes//60

print("The number of days is", days, end=', ')
print("and the number of minutes is", minutes, end='')

minutes = minutes%60

print("the number hours is", hours, end=', ')

input_str = input("Enter a quantity of minutes: ")
hours = hours%24
```

The program consists of correct statements that are in the wrong order. Here is an example of how the program is supposed to behave:

```
Enter a quantity of minutes: 3500
The number of days is 2, the number hours is 10, and the number of
minutes is 20.
```

Download the program and rearrange the statements so that it operates correctly.

HINT: check you understand what the integer operations '//' and '%' do, and think how, given an amount in minutes, you would calculate the equivalent amount of days, hours and minutes.

Submission

Create and submit to the automatic marker a Zip file called `ABCXYZ123.zip` (where ABCXYZ123 is YOUR student number) containing `secret.py`, `secret_2.py`, `temperature.py`, and `time.py.`

END