Programming Assignment 2

Due Sep 29, 2020 by 11:59pm **Points** 8 **Submitting** a file upload

Create a Python file for each of the following programming tasks. The files should be named for the task. For example, if the task is called leapYear, then your file should be called leapYear.py. You should also make sure you include your name at the top of every file in a comment. These are to be submitted via canvas.

Note that some of these require user input. Here is an example of how you may do this

```
name = "Melisa"
age = input("Your age? ")
print("So, you are already " + age + " years old, " + name + "!")
```

```
Your age? 23
So, you are already 23 years old, Melisa!
```

1) increasing (2 points)

Write a function called strictlyIncreasing, with a parameter howMany. Have the user input howMany numbers (i.e as many numbers as the parameter states), print True if they are in strictly increasing order.

This will be multiple user prompts. The first to input how many numbers, and then to have the user input that many inputs (i.e. if they say 5 numbers, then you will loop and have them give you 5 numbers)

This means that b is greater than a and c is greater than b. For example 2 5 11 12 88, or 4 5 6 7 8 are strictly increasing but not 4 6 5 7 8 or 5 5 7 8 9.

2) sentinel (2 points)

Write a program that allows the user to enter number, one at a time. The program will add up the numbers the user entered. When the user enters a -1 the program will finish and print the sum of the numbers the user entered (not including the -1). Hint: a while loop works well. As an example:

```
enter a number (-1 exits):
5
enter a number (-1 exits):
8
enter a number (-1 exits):
3
enter a number (-1 exits):
-1
sum is: 16
```

3. digitsAsWords (2 points)

Write a function called digitsAs Words that takes a 4-digit integer as a parameter and prints its digits as words. For example, if the input value is 3451 then the program prints: three four five one.

Note that your function will have to break the number into its digits and then figure out the wording for each digit. It is often best practice, and cleanest looking to use a helper function to do part of this work. In this case, you might create a function called digitToWord that returns a String representation of a given digit. So for example, digitToWord(5) should return "five" to the caller.

To implement the top-level digitsToWords function, you now simply need to use math (hint: divide and mod) to obtain each digit, one and a time. You can then call your helper function to get the String representation. It isn't required that your strings print all on the same line.

For extra practice, try expanding this to work with any number given rather than always 4 digits.

4. allOccurrences (2 points)

Write a function called allOccurrences that takes a search string and a target string. It should return a list of all the indices where the target string occurs in the search string. It should return an empty list if the target doesn't occur. For example, given the following DNA sequence string, s, it should find all locations where the target sequence "ca" occurs:

```
s = "agctagcatgcatcttactggca"
print(findAllOccurrences(s, "ca"))  # should print out [6, 10, 21]
```

Note this does NOT need to use lists to store the indexes, you may just print them

Hint: use the string method find with the extra parameter to control where it starts the search. You will also need to use a loop as you don't know how many occurrences you will find.

Some Rubric (1)

Criteria	Ratings			Pts
Q1)	**2 pts** **Full Marks**	**1 pts** **Half marks**	**0 pts** **No Marks**	2 pts
Q2)	**2 pts** **Full Marks**	**1 pts** **Half marks**	**0 pts** **No Marks**	2 pts
Q3)	**2 pts** **Full Marks**	**1 pts** **Half marks**	**0 pts** **No Marks**	2 pts
Q4)	**2 pts** **Full Marks**	**1 pts** **Half marks**	**0 pts** **No Marks**	2 pts
				Total Points: 8