THE UNIVERSITY OF THE WEST INDIES
Department of Computing
COMP1127–Introduction to Computing II

Lab 1

Exercise

Type these expressions in the python shell.

```
>>> 'a'+'pple'
>>> ''+'apple'
>>> 4+[5,6]
>>> [4]+[5,6]
>>> ord("a")
>>> chr(97)
>>> ord("a")+5
>>> chr(102)
>>> chr(ord("f")-5)
>>> ord("a")-32
>>> chr(ord("a")-32)
```

All characters on the keyboard have an equivalent ASCII number. They can be converted to a number using the function `ord` and can be converted back to a character using the function `chr`. In the exercise above you have an idea what ASCII numbers of some characters are e.g. "A" is 65 and "Z" is 90 (all other alphabets are in between 65 and 90 in a sequence) and similarly lowercase "a" ASCII equivalent is 97 and "z" is 122.

Problem 1

Write a function `isUpper` which given a character as input returns True if it is uppercase and False otherwise.
Write a function `isLower` which given a character as input returns True if it is lowercase and False otherwise.

```
>>> isupper('A')
True
>>> isupper('a')
False
```

Problem 2

Write a **recursive** function lowercase that converts all characters in a given string to lowercase. Note `ord` of character returns the equivalent ASCII number and if 32 is added to an ASCII number of an uppercase character then it can be converted to a lowercase (`chr` converts a number back to character). If the character is already lowercase then no conversion is required. [Hint: Strings are like lists and hence can be manipulated as lists.]

```
>>> lowercase("Hello")
'hello'
>>> lowercase("HeLlO")
'hello'
```

Problem 3

Write a **recursive** function `encode` that takes a string as an input. For each character in the string add 5 to its `ord` value and then convert it to a character by using `chr`. Also write another function `decode` that takes a string as input and subtracts 33 from each character to return the original string.
[Hint: string are like list hence they can be manipulated like lists]

```
>>> encode("hello")
'mjqqt'
>>> decode("mjqqt")
'hello'

>>> encode("this is a test run")
'ymnx%nx%f%yjxy%wzs'


>>> decode('ymnx%nx%f%yjxy%wzs')
'this is a test run'
```