Programming Assignment: Base 64 Encoding

Introduction

Information often needs to be encoded in order to be safely transmitted across the internet. This information may be text, numbers, symbols, or control characters that do not print. In modern internet browsers, this information is often needed in the website address. One commonly used method for encoding this information is *Base 64 Encoding.*

The Algorithm – Encoding

1. Begin with a message whose length is divisible by 6.
2. Convert the letters in this message to their ASCII code, which is an integer representing a letter, number, or symbol.
   - The command "ord" will convert a letter to its ASCII code.
   - You can "map" this command to the message string to do the conversion. That is, "map(ord,message)"
3. We now convert each number in the list to an 8-bit binary number. That is, a binary number with 8 digits. Although we have already written code for this, let me suggest the following function:

```
def binary8bit(n):
    ans=format(n,'b').zfill(8)
    return(ans)
```

4. Put all of your binary numbers together into one long string.
5. Break the string into lists of length 6.
6. For each list of length 6, convert the binary number into an integer. (We have already written a function in class to do this.)
7. Use this integer as an ASCII code to convert to a letter. Do this for each integer in your list. To do this using a standard coding, use the following:

```
# Use for encoding from integers to letters via ASCII codes

alpha={i:chr(65+i) for i in range(26)}
beta={i+26:chr(97+i) for i in range(26)}
gamma={i+52:chr(48+i) for i in range(10)}
delta={62:'+',63:'/'}
alpha.update(beta)
alpha.update(gamma)
alpha.update(delta)
def text(t):
    return(alpha[t])
```

8. Your function should end by returning the encoded message.

What to turn in

A Sage worksheet containing:

1. Your function
2. Encoding of the message "DiscreteMathematicsIsFun"
3. Encoding of a message of your choice

Hint: To check the correctness of your encoding, use the following:

```
import base64
base64.b64decode(your_encoded_message)
```