

# Programming Assignment 2: Base 32 Encoding

Due October 11

## 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 or for special processing through a JSON file, etc. One commonly used method for encoding this information is *Base 32 Encoding*.

## The Algorithm – Encoding

1. Begin with a message whose length is divisible by 5. (You can add characters, such as ‘x’, at the end of the message to make the length divisible by 5.)
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 5.
6. For each list of length 5, convert the binary number into an integer. (We have already written a function in class to do this.)
7. Use this integer as a code to convert to a letter. Do this for each integer in your list. Base 32 encoding only has  $2^5 = 32$  possible characters. Typically, these are: 0:A, 1:B, 2:C, ..., 25:Z,26:2,27:3,...,31:7. To do this standard coding, use the following code, which you will find in a file in your assignment directory.

```
# Use for encoding from integers to letters via ASCII codes  
  
alpha={i:chr(65+i) for i in range(26)}  
alpha.update({26:'2',27:'3',28:'4',29:'5',30:'6',31:'7'})  
def text(t):  
    return(alpha[t])
```

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

## What to turn in

A Jupyter notebook in the `assign2` folder containing:

1. Your function
2. Encoding of the message “DiscreteMathematicsIsFunx”
3. Encoding of a message of your choice

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

```
import base64  
base64.b32decode(your_encoded_message)
```