

Computer Science 220L

Laboratory 7 – Functions

Learning objectives:

Writing functions.

Directed Activities:

Download `Lab07.py`.

After the completion of each activity, show your answers to an instructor and switch driver and navigator.

1. Checksum Calculator

Most published books have an associated ISBN, a number printed on the back cover of the book and used to uniquely identify the book. The last digit, called the “checksum”, is a check digit used to catch typing errors when the number is entered.

There are many ways to calculate a checksum depending on its use. With 10 digit ISBNs, the method is to take each digit of the number and multiply it by its positional value. (The leftmost digit is considered position 10 and the rightmost digit is considered position 1.) The sum of these products mod 11 should be 0. The rightmost digit, called the “checksum”, helps ensure that the preceding nine digits are correct.

Example: Consider the ISBN number

$$0072946520 = 0*10 + 0*9 + 7*8 + 2*7 + 9*6 + 4*5 + 6*4 + 5*3 + 2*2 + 0*1 = 187$$

Write a function `calcChecksum(isbn) → int` that accepts as a parameter a string representing an ISBN and returns the checksum as an int. Add code to `main()` to test your function. Wouldn't it be nice if we could actually see whether this checksum checks out? More on that next lab...

2. Send a message to a friend

Write a void function `sendMessage(file, friend)` that takes the name of an existing textfile and the name of a friend as parameters. This function should read the specified textfile and write the contents to a new textfile named after your friend, e.g. `bob.txt`. You might imagine we could do some Python magic to email the message to a friend, but we'll keep it safe and responsible by simply writing to a new local file instead.

3. Send a secret message to a friend

Copy your `sendMessage(file, friend)` function above to a new function called `sendSafeMessage(file, friend, key)`. In the `Lab 7.py` file is a working `encode()` function from last week. You can use this function or replace it with your `encode()` that you wrote last week.

`sendSafeMessage(file, friend, key)` should make a call to the `encode()` method so that your file contents are encoded before being written out. You'll have to modify your `encode` method so that it doesn't ask the user for the message or the key but rather accepts them as arguments from the caller function `sendSafeMessage()`.

4. Put the encryption function in a separate file and call it.

Your safe message was great, but just one more thing...It seems like it would be better to keep your encryption methods safe and secure in a separate file. If we wanted to, we could hide the file in some secret directory, but for now just create a new file in the same directory you're working in and move your encode() function there. Modify the following example import statement so that it imports your new safe and secure encode method:

```
from <fileNameWithoutSuffix> import encode
```

Demonstrate that sendSafeMessage() still works with the imported encode function.

5. Send an uncrackable message to a friend.

You've learned some better ways to encode messages than just using a static key...make a copy of your sendSafeMessage() function and call it sendUncrackableMessage(file, friend, pad).

In your encodeBetter() function (a working function is included), you used a string message to shift the value of each character in your message by a different value. For our uncrackable message, we'll create a one-time pad at <https://www.fourmilab.ch/onetime/otpjs.html>. Change the 'number of keys' option to 1, and change the 'key length' to be more characters than your message. Ideally, we'd have the same number of characters, so feel free to count the characters in your message to get it exact if you want...

You should copy the pad that is generated to a text file and pass this text file as the 'pad' argument. The string pad should be used in the same way you used the cipher string in your encodeBetter() method. Modify encodeBetter() to work with your code.

Go ahead and put this encryption method in your separate encryption file and be sure this still works with the encode function in a safe place.

Upload the files to both student accounts.