Work with your neighbor. (This will be graded for participation only.)

1. Hashing (for review). Answer the following questions:

   a) How do we guarantee that every slot will be visited when using double hashing?

   ANS: You must make sure that the probe function returns values that relatively prime to the hash table size.

   b) Define load factor. What is the equation for the load factor of a hash table?

   ANS: A measure of how full a hash table is. The equation is N/M, where N is the number of keys and M in the hash table size.

   c) What is the von Mises Birthday Paradox?

   ANS: If there are 23 or more people in the room, there is a greater than 50% chance that two people will have the same birthday.

2. The hash function `hash(key, M)` below takes a string key and a table size `M` and computes the hash value by summing the ord of the characters mod `M`. (Recall that `ord(c)` returns the ASCII value of `c` in decimal.) Modify `hash()` to incorporate the position of the character before including it in the sum:

```
def hash(key, M):
    sum = 0
    for c in key:        # key is a string
        sum += ord(c)
    return sum % M
```

   ANS:

```
def hash(key, M):
    sum = 0
    for i in range(len(key)):
        sum += ord(key[i])* i
    return sum % M
```

3. (Debugging) The following code has many errors, packed into a very little space: syntax errors, runtime errors that crash the program, and even some logical errors which cause silent bugs. How many can your group find?

```
import Random
a = random.randint(1,12)
b = random.randint(1,12)
for i in range(l0):
    question = "What is "+a+" x "+b+"? "
    answer = input(question)
    if answer = a*b
        print (Well done!)
    else:
        print("No.")
```

Make a list of the bugs you found.

ANS:
`Random` should not be capitalized

Need to cast `a` and `b` to `str` in the question

Need to convert the answer to an integer for the comparison

Need quotes around `Well done!`

Need to either break out of the loop when the user succeeds - or else generate new `a`, `b` every time. (The purpose of the program isn't 100% clear.)

NOTE: Problems 4 & 5 have been moved to ICA-37.