Cryptography Challenge #2

In the previous challenge, you successfully decrypted the pirates' secret location, allowing Tanzania's military to apprehend them. Your skills have made you a hero in Zanzibar, where the people are able to return to their livelihoods without fear of being robbed or held captive.

After learning of your success, Tanzania's ambassador to Somalia asks to meet with you about a confidential matter. He tells you that when he was in Somalia's capital city yesterday, a brother of one of the pirates came to him with some interesting information. The man said that his brother had been responsible for hiding the pirates' stolen money; each month, he would move the money to a new location, then tell the other pirates about the new hiding place. Of course, he would *encrypt* the location before giving it to the others so as to prevent anyone else from finding the money.

After the pirate was arrested, his brother went to his house and found this on a piece of paper:

D	E	j	L	S	\x1e	;	$	\x10	9	2	\x12	8	?	\x08	=	?	\n	0	>	\x16	9	=	\x1c

The brother is certain that this is the encrypted current location of the pirates' money, and that the pirate used this Python program to perform the encryption:

```
def encrypt(secret_message, key):
    key = (key * (len(secret_message) / len(key) + 1))[:len(secret_message)]
    assert len(secret_message) == len(key)
    return [chr(ord(a) ^ ord(b)) for a, b in zip(secret_message, key)]

def decrypt(encrypted_message, key):
    key = (key * (len(encrypted_message) / len(key) + 1))[:len(encrypted_message)]
    assert len(encrypted_message) == len(key)
    return ''.join([chr(ord(a) ^ ord(b)) for a, b in zip(encrypted_message, key)])


secret_key = '[redacted]'
secret_location_of_money = '[redacted]'

encrypted_location_of_money = encrypt(secret_location_of_money, secret_key)
print "Encrypted location of money is: %s" % encrypted_location_of_money

decrypted_location_of_money = decrypt(encrypted_location_of_money, secret_key)
print "Decrypted location of money is: %s" % decrypted_location_of_money
```

The ambassador gives this program to you, and you notice that it is very similar to how the pirates hid their previous location: they are XORing the ASCII characters of the message with the ASCII characters of their secret key. But this time they're using a much shorter secret key, which they repeat over and over until it's as long as the message, before performing the XOR.

You also notice that now both the secret key *and* the secret location of the money are blacked out in the pirates' program. All you know is their encrypted location and what the pirates' brother says is true:

- the secret key is 3 ASCII characters long
- the decrypted message starts with "**MONEY:**"

The ambassador wants to know if you can write a Python program that will find the secret key revealing the location of the pirates' money. If you succeed, Tanzania will be able to recover the millions of shillings the pirates stole and return them to the people of Zanzibar.

Before you write any code, first **think** about what you already know, and how you can use it to find the pirates' secret key. Answer each of the questions below in the order that they appear.

1) The pirate's brother says that the key is made up of 3 ASCII characters. How many different ASCII characters are there? Look at an ASCII table if you don't remember.

   ____

2) Using your answer for #1 and the fact that the pirates' key is 3 characters long, how many possible keys are there? To help you, think about how we can calculate how many different bytes exist: a byte contains 8 bits, and each bit has 2 different values (0 or 1). So there are:

   2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 2^8 = 256 different bytes

   ______________________________ = ____ = ____________________ different keys

3) Write a Python program that prints out every possible 3-character ASCII key. The program should print each key on a separate line, and it should print out the same number of keys as your answer for #2 above. As you write your program, remember that you can use Python's *ord(char)* function to get the decimal value of an ASCII character, and the *chr(num)* function to get the ASCII character for a decimal value. Also think about what happens when loops are nested together, and whether or not that might be useful to you in your program.

4) Using your program from #3, change it so that it calls the pirates' *decrypt(encrypted_message, key)* function for every 3-character ASCII key that your program generates. Give each possible key, along with the pirates' encrypted location, to the function, then store the resulting decrypted message in a variable. Check each decrypted message to find the one that starts with "**MONEY:**" - when you've found it, you know that you've successfully found both the pirates' key and the location of the pirates' money. Your program should print out both the key and the decrypted location, then exit. A few things to keep in mind:

   - Use this variable in your Python code to hold the pirates' encrypted message:
     ```
     encrypted_location = ''.join(['D', 'E', 'j', 'L', 'S', '\x1e', ';',
     '$', '\x10', '9', '2', '\x12', '8', '?', '\x08', '=', '?', '\n', '0',
     '>', '\x16', '9', '=', '\x1c'])
     ```
   - To find out if a string starts with another string, use the Python *startswith(str)* method.
   - When you print out the pirates' key, do so in hexadecimal. To do so, use the Python *encode('hex')* method.
   - To exit a Python program early, use *exit(0)*.

5) Bring your answers to these questions, along with the pirates' key and the location of their money, to your ICT Practical instructor. Enter the GPS coordinates into Google Maps and verify with your instructor that the location you found is correct.