Name: Michael Schwamborn Date:

1. Write a program that will ask the user for the offset and message. Using the following input, answer the following questions:

**Encode the following test cases:**

1. Caesar cipher with shift +3 **hello Mr Sabo KHOOR PU VDER**
2. Caesar cipher with shift +12 **Hazen Rocks TMLQZ DAOWE**

**Decode the following messages:**

1. Caesar cipher with shift +21 **adiyevhznwjiy FINDJAMESBOND**
2. Caesar cipher with shift +24 **ncwrmlkyllgle PEYTONMANNING**

**What is the offset: Use frequency analysis.**

1. Caesar cipher using frequency analysis. Shift is 6 **kbkxeutk everyone**
2. Caesar cipher using frequency analysis. Shift is 11 **espntaspcsldmppymczvpy thecipherhasbeenbroken**
3. Caesar cipher using frequency analysis. Shift is 6 **kgyezuhxkgq easytobreak**
4. Write up a paragraph response to, “Consider designing a more sophisticated algorithm that would be more secure than the Caesar cipher. What sequence of steps could you perform to securely encode your message that would make it harder for a hacker to crack your code?”

I would add something to my cipher where the code would give something to help decode the rest of the message. It would be similar to salts in hashing but the answer to how to solve it would be in the message. For example, in the message there would be string of numbers that were put into a format like a date but really it is a list of steps for how to move the cipher.

<http://www.simonsingh.net/The_Black_Chamber/caesar.html> - site is where you can check your work.

1. Explain the difference between the Caesar Cipher and the Vigenere Cipher?

The Caesar Cipher is a simple cipher that moves letters down by an offset and it can be easily found out. The Vigenere cipher uses a cipher word in addition to the message. The two strings are put on a rectangle where each character is lined up to each other in an alphabet that is being shifted over every new line.

**package** test;

**public** **class** **CeaserCipher** {

**public** **static** **void** **main**(**String**[] args) {

}

**public** **static** **String** **encrypt**(**String** word, **int** offset) {

word = word.toUpperCase();

**String** **encryptedWord** = "";

**for**(**int** **i** =0; i<word.length();i++) {

**if**(word.charAt(i) == ' ')

encryptedWord += " ";

**else**

encryptedWord += (**char**) ((word.charAt(i) - 65 + offset)%26 + 65);

}

**return** encryptedWord;

}

**public** **static** **String** **decrypt**(**String** word, **int** offset) {

**String** **dencryptedWord** = "";

**for**(**int** **i** =0; i<word.length();i++) {

**if**(word.charAt(i) == ' ')

dencryptedWord += " ";

**else**

dencryptedWord += (**char**) ((word.charAt(i) - 65 + (26-(offset%26)))%26 + 65);

}

**return** dencryptedWord;

}

}