Programming Assignment 1

(Note that programming assignments are only required in order to pass with distinction.)

Write a program that allows you to "crack" ciphertexts generated using a Vigenere-like cipher, where byte-wise XOR is used instead of addition modulo 26.

Specifically, the ciphertext



was generated by encrypting English-language text using the following C program:

#include <stdio.h>

#define KEY\_LENGTH 2 // Can be anything from 1 to 13

main(){

unsigned char ch;

FILE \*fpIn, \*fpOut;

int i;

unsigned char key[KEY\_LENGTH] = {0x00, 0x00};

/\* of course, I did not use the all-0s key to encrypt \*/

fpIn = fopen("ptext.txt", "r");

fpOut = fopen("ctext.txt", "w");

i=0;

while (fscanf(fpIn, "%c", &ch) != EOF) {

/\* avoid encrypting newline characters \*/

/\* In a "real-world" implementation of the Vigenere cipher,

every ASCII character in the plaintext would be encrypted.

However, I want to avoid encrypting newlines here because

it makes recovering the plaintext slightly more difficult... \*/

/\* ...and my goal is not to create "production-quality" code =) \*/

if (ch!='\n') {

fprintf(fpOut, "%02X", ch ^ key[i % KEY\_LENGTH]); // ^ is logical XOR

i++;

}

}

fclose(fpIn);

fclose(fpOut);

return;

}

(Of course, when encrypting I used a random key length and chose each byte of the key at random.) The plaintext contains upper- and lower-case letters, punctuation, and spaces, but no numbers.

Recovered the original plaintext,