Ricardo Ruiz Rodriguez

**Week One Homework Assignment**

**1.** Shift of 16

Plain Text: welcome\_to\_virtual\_cryptography.

**2.** Shift of 8

Plain Text: mathematics\_and\_considering\_all\_possibilities\_will\_be\_important\_in\_this\_course.

**3.** Decryption Keys used:

a=7

b=24

Plain Text: though\_the\_affine\_cipher\_is\_easy\_to\_break\_with\_a\_computer\_it\_allows\_me\_to\_introduce\_you\_to\_quite\_a\_few\_important\_ideas\_in\_cryptography\_such\_as\_the\_extended\_euclidean\_algorithm.

**4.** Ciphertext: kwjiqgkqcygxuaztwdwregyszaqjqlhqoyztwztwswvvztyvyzzylsajwpqquaztwfqbmalwffyqlngvzgly

**Decryption:** For both ciphers, affine and shift, I wrote a brute force method for decryption in which the program loops through all possible decryption keys, prints the plain text and then I looked through the output for any legible text that looked to be correct.

**Encryption:** For the affine cipher encryption I wrote a c program to aid in encrypting the text. The program takes the desired text, a, and b keys. It then loops through the text and prints out each encrypted character using the formula (((a \* (char – ‘a’)) + b) % 26 + ‘a’.

**Shift Cipher Snippet( C ):**

// LOOP THROUGH ALL POSSIBLE SHIFTS 1-26

for(int i = 1; i < 26; i++){

// Print Shift #

printf("Shift %d: ", i);

// Print decrypted text

for(int j = 0; j < size; j++){

printf("%c", (cipher[j] - 'a' - i + 26) % 26 + 'a');

}

printf("\n\n");

}

**Affine Cipher Snippet( C ):**

// Affine Cipher

// Encryption and Decrption

#include <stdio.h>

#include <string.h>

// VALID a FOR DECRYPTION

int valid[] = {1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25};

int main(void){

// Get Code for Operation 1: Encryption 2: Decryption

int code;

scanf("%d", &code);

// Get Text

char text[1000];

scanf("%s", text);

// Get size of word

int size = strlen(text);

//ENCRYPTION

if (code == 1){

// Get Desired Encryption Keys

printf("Enter Key: (a b)");

int a, b;

scanf("%d %d", &a, &b);

// Print Cipher Text

for (int i = 0; i < size; i++){

printf("%c", ((a \* (text[i] - 'a')) + b) % 26 + 'a');

}

}

// DECRYPTION

else if (code == 2){

// Loop Through all possible a decryption keys

for (int i = 0; i < 12; i++){

//Loop through all b possibilities (0 - 26)

for (int j = 0; j <= 26; j++){

// Print Current a and b for decryption

printf("a: %d b: %d - ", valid[i], j);

// Loop through Cipher Text

for (int k = 0; k < size; k++){

// Print Plain Text

printf("%c", ((valid[i] \* (text[k] - 'a')) + j) % 26 + 'a');

}

printf("\n");

}

printf("\n");

}

}

return 0;

}