Roy Basmacier 24813

Question 1)

1. I will use brute force to find the 2 words it could correspond to shift each character by i also take the modules of 26 so it loops around the alphabet. i=(0,1,2,...,25)
2. We get the following:
3. [('NZWO', 0), ('OAXP', 1), ('PBYQ', 2), ('QCZR', 3), ('RDAS', 4), ('SEBT', 5), ('TFCU', 6), ('UGDV', 7), ('VHEW', 8), ('WIFX', 9), ('XJGY', 10), ('YKHZ', 11), ('ZLIA', 12), ('AMJB', 13), ('BNKC', 14), ('COLD', 15), ('DPME', 16), ('EQNF', 17), ('FROG', 18), ('GSPH', 19), ('HTQI', 20), ('IURJ', 21), ('JVSK', 22), ('KWTL', 23), ('LXUM', 24), ('MYVN', 25)]
4. From the result we can conclude that the possible words can be either COLD or FROG with the keys 15 or 18 respectively

Question 2)

1. Since we are given the most frequent letter "a" we can find out what "a" corresponds to in the ciphertext
2. We can see that the most frequent letter in the cipher text is "h"
3. We have the following plaintext-ciphertext pair ("a", "h") -> (0,7)
4. 7 = 0x(alpha) + beta (mod 26) => we know beta is 7
5. The gcd of alpha and 26 must be one in this cipher
6. Now we I will use brute force to find alpha
7. x = gamma . y + theta (mod 26)
8. From the text we can see that the only one that makes sense is the following
9. A successful man is one who can lay a firm foundation with the bricks others have thrown at him.
10. with the key as (23, 7, 17, 11) = (alpha, beta, gamma, theta)

Question 3)

1. Since we have 31 characters in our language and we want to encrypt two letters at a time, we will calculate the possible combinations of each character which is 31x31 = 961
2. x and y $\in Z_{961}$
3. Key: k = (α, β) and α, β $\in Z_{961}$
4. Encryption: Ek(x) = y = α · x + β mod 961
5. Decryption: Dk(x) = x = α^−1 · y + γ mod 961
6. Key Space:
7. β can be any number in Z961.
8. gcd(α, 961) = 1 → α ∈ A → len(A) = phi(961) = 930
9. The key space has 961 · 930 = 893730.

Question 4)

1. I modified the affine algorithm
2. We will define the bigram language and its inverse by finding all the permutations of the letters of length 2

3. The Affine Algorithm suggest that alpha can only be the numbers relatively prime to the size of the (bigram) language
4. I used a python library called enchant which helps distinguish english words
5. I will store all plain text which have 5 or more english words in them by splitting the text where there are spaces and checking if the words are english
6. After a couple of minutes of computation, we raise our hands
7. plaintext:
8. THOSE WHO BELIEVE IN TELEKINETICS, RAISE MY HAND..
9. key: (626, 843) = (gamma, theta)

Question 5)

1. I implemented my Vigenere Algorithm
2. What is the plain text?
3. I REFUSE TO ANSWER THAT QUESTION ON THE GROUNDS THAT I DON'T KNOW THE ANSWER.

Question 6)

1. We must find the length of the key in order to decipher the text
2. First we must shift the cipher text and count the number of coincidences
3. Increment the shift amount
4. If we exceed a predetermined shift amount we continue else we continue Incrementing the shift
5. Let's see which shift has the most coincidences
6. It looks like the length of our key is 7 since we have the most coincidences when we shift it 7 times
7. So there is 7 shift ciphers we have to crack
8. First we partition the cipher text into 7 sub texts
9. # Now for each sub text we will apply the frequency analysis
10. # so we have 7^len(possibleKeys) amount of possible keys
11. # test all permutations of the possible shifts

key: KLAWISZ

plaintext:

Whose woods these are I think I know.
His house is in the village, though;
He will not see me stopping here
To watch his woods fill up with snow.
My little horse must think it queer
To stop without a farmhouse near
Between the woods and frozen lake
The darkest evening of the year.

He gives his harness bells a shake
To ask if there is some mistake.

The only other sound's the sweep
Of easy wind and downy flake.
The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.