Eder Mazariegos

CIS3362

09/24/17

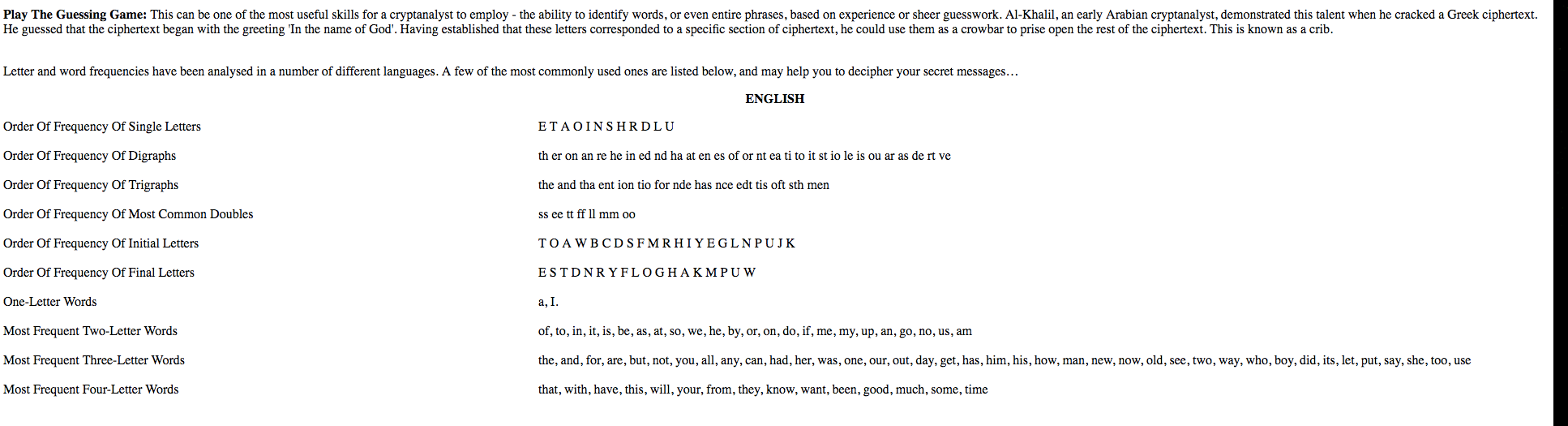
Homework 3

1. Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

* For this cipher decryption, I thought of making a program that would use the index of coincidence to test possible key solutions to crack the cipher, but that didn’t seem to work out how I had planned, so I instead took another approach. To crack the code, I used the crypto tool we were provided with on the class website (The html one, not java) and I only used it for getting the letter frequency within the cipher text and with that data, I stored it into a chart that held the Order of Frequency Single Letters. The chart looked something like this…

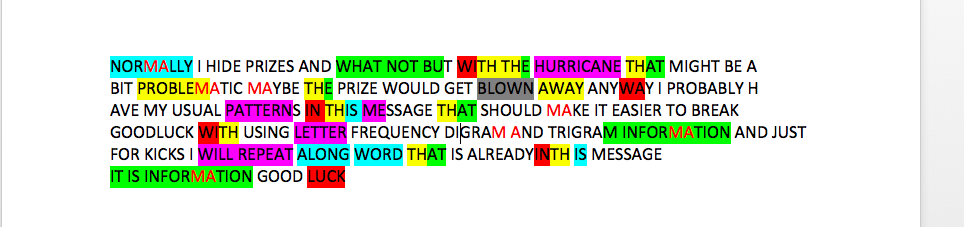
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E | T | A | O | I | N | S | H | R | D | L | U |
| P | H | C | K | A | R | F | E | S | N | J | Z |



This helped a lot and I remembered the method we learned in class which was simply if it has a higher frequency of occurrence within the text, we pair it with the higher frequencies of letters within the English language. After I did this, I simply opened up Word and replaced some, but not all of the letters, I looked for things that could possibly make sense, but I soon realized that it wasn’t working the way I expected and I needed more information, so I looked up the Order Of Frequency Trigraph (the, and, tha, ent, ion, tio, for, nde, has, nce, edt, tis, oft, sth, men). This helped to narrow down the search and give real outputs and things that made sense. When I swapped the letters for the two that occurred the most in pair (“hn”) and replaced it with T and H which were in the chart above and in somewhat close or near proximity to something that could match up, I found that the word “The” was forming. It took me a lot of trial and error and a lot of hours with hacking away at it, but when I found this 

It sped things up a bit. I included screenshots of every vital move I made and every move that formed some type of words and when you look through them in chronologic order (Times are not right because I re-did some of the processes, simply to show screenshots of what I did), you’ll see that the steps got me to very good points within the decryption process and highlighting helped immensely when I needed to keep track of every move I was making. I attached the screenshots with a file entitled “Sub Cipher 1” and it illustrates in more detail how I moved through the entire process. Also, once I found a word, I simply replace every letter with the letter that corresponded to the correct letter in that word in uppercase so I knew where it was and what other words were coming about or changing.

DECODED MESSAGE:



KEY:

Cipher Plain

|  |  |
| --- | --- |
| A | r |
| B | j |
| C | i |
| D | b |
| E | s |
| F | n |
| G | w |
| H | t |
| I | c |
| J | m |
| K | e |
| L | f |
| M | d |
| N | h |
| O | u |
| P | a |
| Q | x |
| R | o |
| S | l |
| T | z |
| U | v |
| V | k |
| W | q |
| X | p |
| Y | y |
| Z | g |

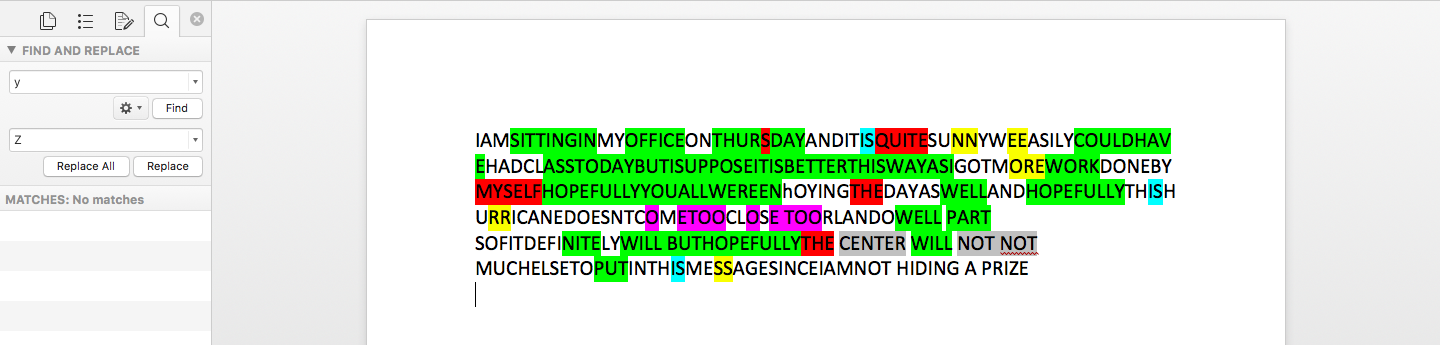
1. Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

* For this problem, I used the same method as above, except for the change in the frequency chart I made which looked like this

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E | R | U | G | B | X | L | K | T | J | A | F |
| E | T | A | O | I | N | S | H | R | D | L | U |

With the decryption of this cipher text, it was more difficult and I believe it was that way because of the way words were positioned in the beginning of the cipher text, and for some reason things did not make as much sense as fast (The word that helped the most was hopefully though). However, I felt like this decryption was less annoying because I had already come up with a system that I knew worked. Just like the one above, I have included screenshots of every move I made as I feel that it gives more insight into how I moved rather than me trying to write out another explanation two days after already finishing the decryption…

DECODED:



KEY:

Cipher Plain

|  |  |
| --- | --- |
| A | h |
| B | l |
| C | b |
| D | k |
| E | n |
| F | d |
| G | o |
| H | j |
| I | i |
| J | u |
| K | a |
| L | n |
| M | w |
| N | f |
| O | c |
| P | p |
| Q | r |
| R | i |
| S | g |
| T | y |
| U | t |
| V | v |
| W | m |
| X | s |
| Y | z |
| Z | z |

1. Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

* For this problem, I was forced to use a program to crack the code. I believe the hardest part of this problem was finding a way to decrypt the cipher text without a given key or any hint to what that key might be, and then writing up the code for it. In the end, I drew a layout of how I would go about finding the word that was used for encryption and using that to decrypt the message. I remember in class there was something said about using common words within the English language, so I did just that…. all 20,000 of them. I simply brute forced attacked the cipher text with a file or dictionary of common words use in the English language (File included in folder) and used a formula that would normally be used for the decryption when given a key. However, I found that it still didn’t work and I ran into a lot of problems such as, sifting through all possible outcomes, debugging, and verifying if my code even worked sufficiently. Luckily, I found that sifting through a file of 500 common words at a time to find common words within the decoded messages was far easier than all 20,000 at once and with my own cipher text and key test, I could verify that the code was working sufficiently and debug where necessary. All in all, this code took a long time and I believe that is because I am only one person working this (It felt as though it was almost impossible) and even with all the work I had put in, I didn’t get the correct answer. I got very close though. I found that the word “RANDOM” was the only key that had words within it that made sense and I thought there would be more to it, so I tried things like “RANDOMNESS,” “RANDOMTIMES,” “RANDOMTEXT,” and then of course “RANDOMWORD.”

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

//#define BUFFSIZE 30

//char charbuff[BUFFSIZE], ch, \*c, dispense[10], string[10];

int decrypt(const char \*cipher,const char \*k)

{

int length = strlen(cipher);

int keylength = strlen(k);

char plain[length];

char key[keylength];

strncpy(plain, cipher, length + 1);

strncpy(key, k, keylength + 1);

//Move through and over the original string...

int i, j;

for(i = j = 0; i < length; i++, j++)

{

//decryption for given character

if(!isspace(plain[i])){

char p = ((toupper(plain[i]) - 65) - (toupper(key[(j % keylength)]) - 65));

plain[j] = (p<0?p+26:p) % 26 + 65;

}

//step-back the ciphertext index space

else

j--;

}

//null-terminator

plain[j] = '\0';

printf("%s\n", plain);

return 0;

}

int main(int argc, char \*argv[])

{

int m=0;

char \*string[100];

char line[100];

for (int d = 1; d < 2; d++){

char\* fname = argv[d];

FILE\* file = fopen(argv[d], "r");

if (file == NULL) {

fprintf(stderr, "ERROR cannot find file!\n");

}

while(fgets(line, sizeof line, file)!=NULL){

printf ("%s\n", line);

decrypt("nenusyegjlegnpwzealpffgzcohojvvsjkwoddirsaoomyaoevzwvoztwjvwfsxldyuselxmngoksvzfyifwcaxouevcnxgqpvrwjtbumuofvdcllusmhzpletrusepwejrtgkshafpovafwmaqocojhbxzpccjhvizlhmpwfqxazhcdnsrgkhrfvmlhvuzngksokrvefoxdgkkinhclxqcietprlaehfqwrzqxtulgfdwjzrsrqqdudkhuuflbspvvyzgrqsdaqzsyeelvalsprlwrusmxzvwfuglzuvsdxkunowzzsorwcblboervqtebuamupvbfusrizzoihgenwspscigkhnwweebjbecjlhtpvvnvyjrfphsejkhrlhtafndphbsskkixhktbusmzhyhdefvosapvifrrwvqdcdhnoenweziv", line);

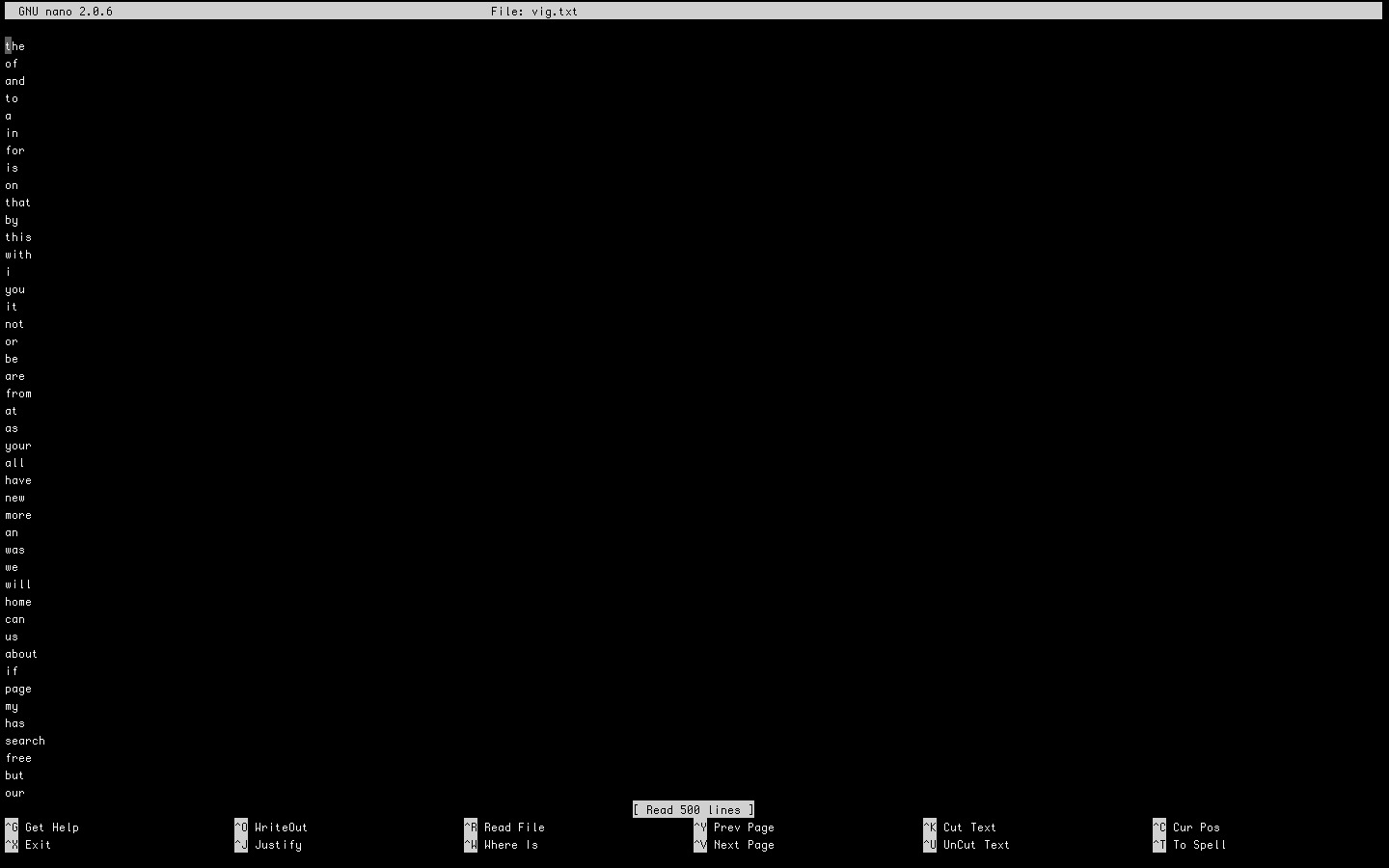
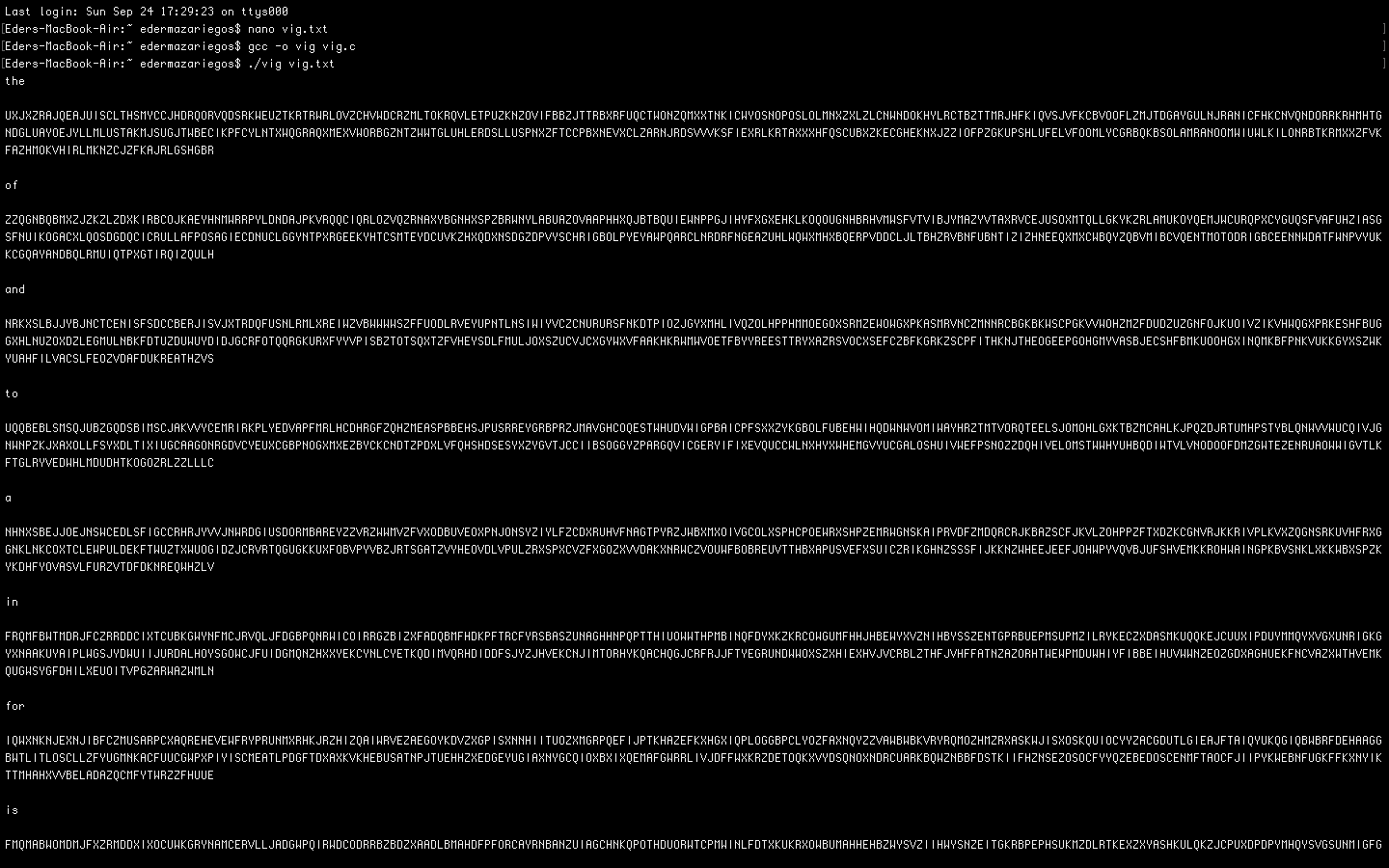
printf("\n");

}

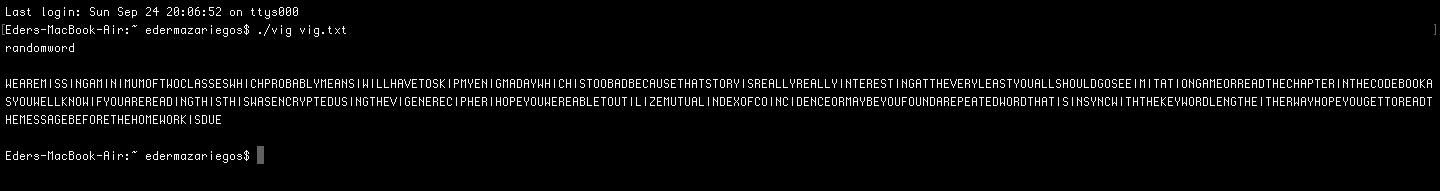
}

}

SCREENSHOTS:



DECRYPTED MESSAGE:

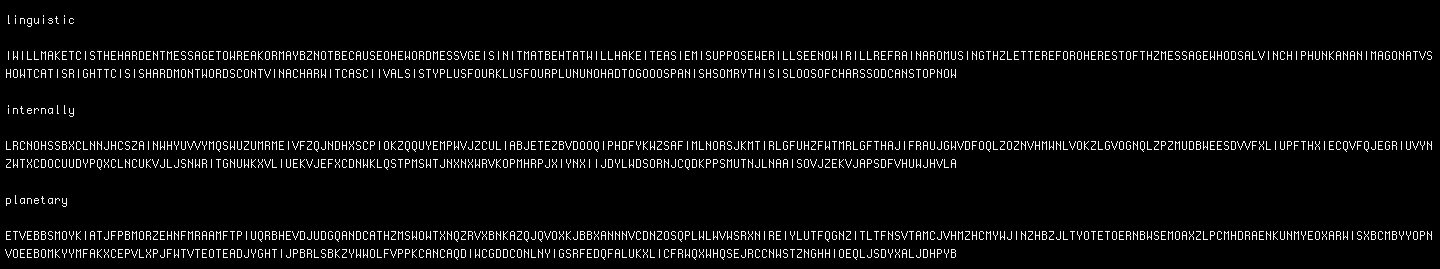


WEAREMISSINGAMINIMUMOFTWOCLASSESWHICHPROBABLYMEANSIWILLHAVETOSKIPMYENIGMADAYWHICHISTOOBADBECAUSETHATSTORYISREALLYREALLYINTERESTINGATTHEVERYLEASTYOUALLSHOULDGOSEEIMITATIONGAMEORREADTHECHAPTERINTHECODEBOOKASYOUWELLKNOWIFYOUAREREADINGTHISTHISWASENCRYPTEDUSINGTHEVIGENERECIPHERIHOPEYOUWEREABLETOUTILIZEMUTUALINDEXOFCOINCIDENCEORMAYBEYOUFOUNDAREPEATEDWORDTHATISINSYNCWITHTHEKEYWORDLENGTHEITHERWAYHOPEYOUGETTOREADTHEMESSAGEBEFORETHEHOMEWORKISDUE

I also have the file with the 20,000 words attached in a folder with this submission. It was just scanned (parsed) through line by line and then ran through the decrypt function above.

1. Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

* For this problem it was a lot easier because the above program was already finished and even though it took twice as long to actually sift through all of the given outcomes and some words kind of did make other words, there weren’t any that made consistent words that could form sentences and I did try other variations of some words in order to find the key because I thought it would be just like the problem above, but none of them made any sense and I felt as though I was getting nowhere. Also, I talked myself out of trying anymore because repeating the same thing repeatedly just seemed insane. In the end, at word 12,181 (I eventually just started from the middle and cut my time in half with the search) I found “LINGUSITIC” and this was the given message….



The code used is the same as the one given above, with the exception that the cipher text hardcoded within was changed.