Final Project Report

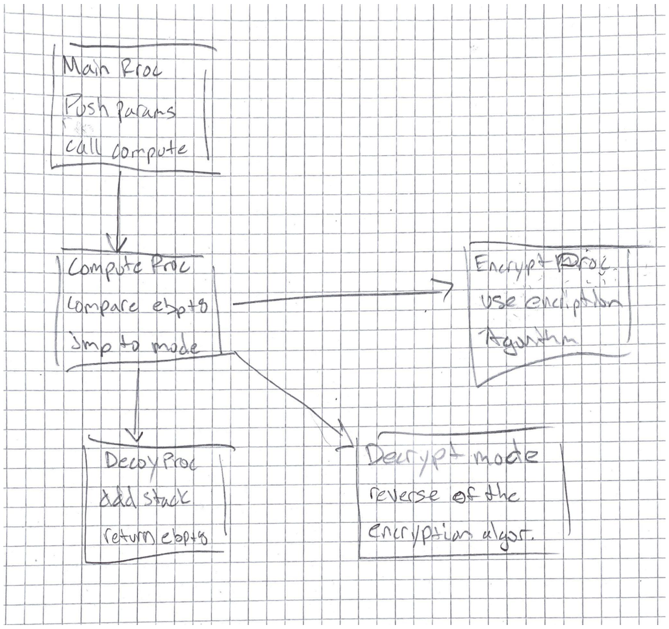
By: Michael Smith

This project proved to be very difficult for me and I was not able to finish it on time. Not being able to use the main or .data sections was something I struggled to overcome while I was trying to complete this program. I do believe I started off strong for this assignment but was not able to implement the algorithms and/or the plans I made as It got closer to the due date. The only extra credit I did was a comment greeting for a TA.

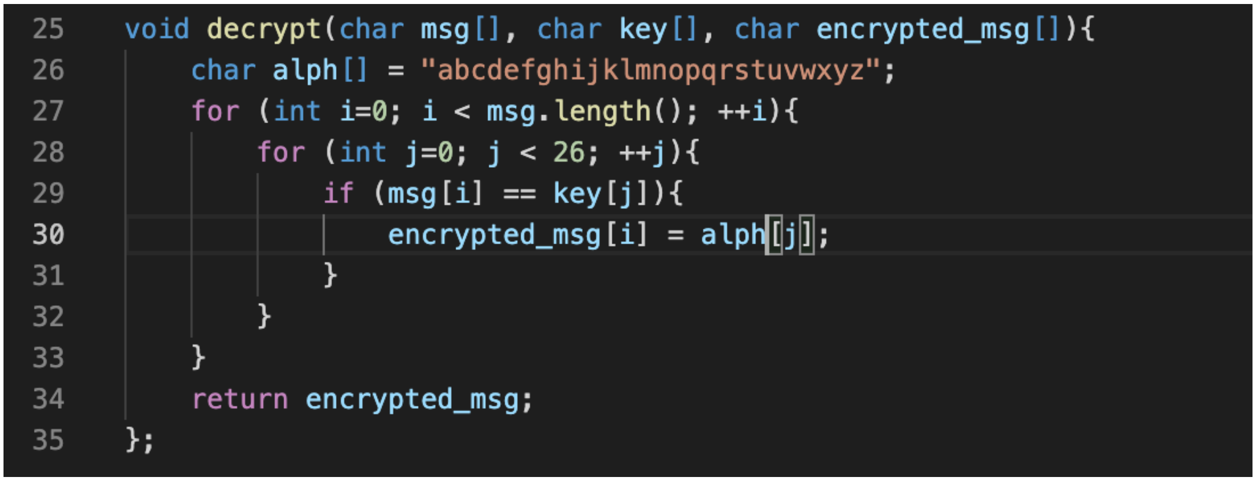
While working on this project I ran into a lot of difficulties and I found most of my help from one of three places: StackOverflow, the course Discord, and the Textbook. For future students of this course, I would highly recommend keeping up to date with the course Discord messages because it is very likely that another classmate is having the same issues as you and the TAs do a great job at answering them. The biggest challenges I ran into mainly involved keeping track of the stack, specifically stack parameters. Pushing from main into compute and then pushing from compute into another sub procedure made it very difficult for me to understand what was going on with the stack as my program was compiling. This made me run out of general purpose registers and I ultimately couldn't find a way to work around this. Looking back, I believe I should have implemented local variables sooner rather than later because I ran out of time and did not have a working program. Below are some diagrams and other bullet points regarding how I approached this project.

How I broke up my procedures:

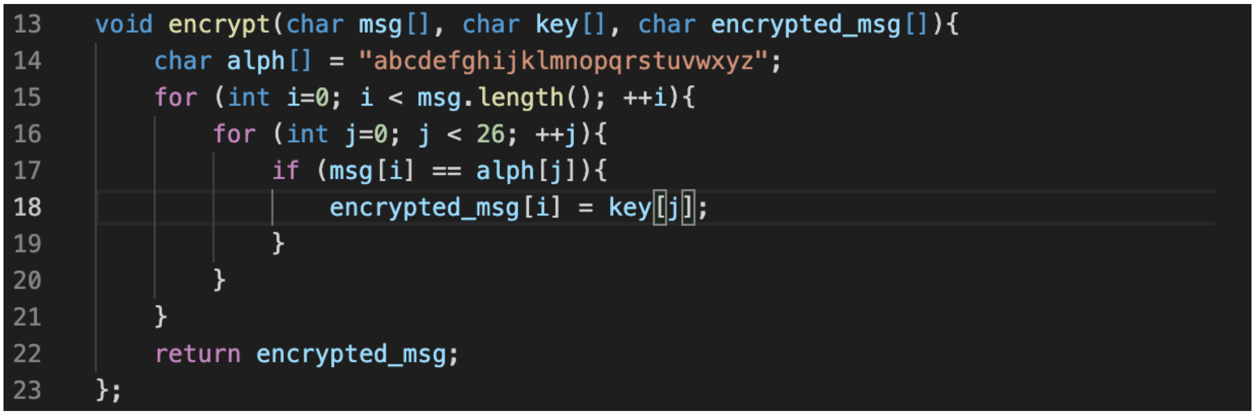
- My original plan was to have four procedures, them being compute, decoy_mode, encrypt_mode, and decrypt_mode.
- I planned on potentially having 1 or 2 other procedures to assist the decryption/encryption algorithms if it proved useful but ultimately I didn't
- Visual Diagram:



- Here is some example C code I wrote to implement my algorithm for the substitution cipher:



```
void decrypt(char msg[], char key[], char encrypted_msg[]){
    char alph[] = "abcdefghijklmnopqrstuvwxyz";
    for (int i=0; i < msg.length(); ++i){
        for (int j=0; j < 26; ++j){
            if (msg[i] == key[j]){
                encrypted_msg[i] = alph[j];
            }
        }
    }
    return encrypted_msg;
};
```



```
void encrypt(char msg[], char key[], char encrypted_msg[]){
    char alph[] = "abcdefghijklmnopqrstuvwxyz";
    for (int i=0; i < msg.length(); ++i){
        for (int j=0; j < 26; ++j){
            if (msg[i] == alph[j]){
                encrypted_msg[i] = key[j];
            }
        }
    }
    return encrypted_msg;
};
```