WRITE YOUR ALL DETAILS HERE ON THIS PAGE LIKE YOU NAME, UNIVERSITY ID AND THE ASSINGMENT NUMBER etc.

Design of Enigma

The Enigma machine was an electromechanical device used for encryption and decryption during World War II. It was first developed in the early 1920s and was used extensively by the Germans during the war. The machine had various versions, including the software version, which had five major parts: the plugboard, the right rotor, the middle rotor, the left rotor, and the reflector. In this answer, we will explain in detail each of these parts and their role in the encryption process.

Plugboard Substitution:

The plugboard is the first part of the Enigma machine, where the plain text given by the user is substituted with respective characters on the plugboard. The plugboard substitutes only uppercase letters. Therefore, all input by the user is converted to uppercase letters, making it easy to encrypt and decrypt text with the same algorithm. The plugboard substitution table shows the respective substitution for every character. The substitution is based on the plugboard settings, which the user could set according to their preferences. The plugboard settings would change the substitution table and, consequently, the encrypted text.

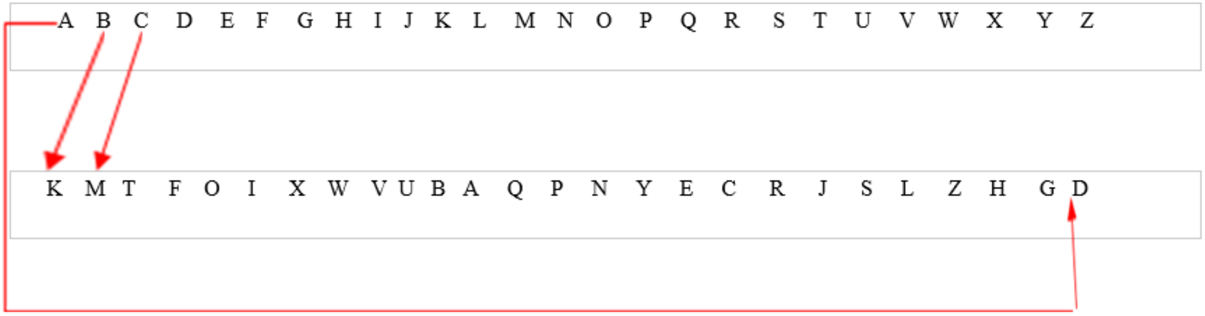
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	P	H	N	M	S	W	C	I	Y	T	Q	E	D	O	B	L	R	F	K	U	V	G	X	J	A

Rotors Substitution:

The next substitution is done on the rotors. The Enigma machine had three rotors, and each rotor had 26 letters on it. The rotor rotates every time a letter is encrypted, changing the substitution table and making it difficult to decrypt. The rotor had two sides, and the first three substitutions were done on one side, and the other three on the backside. When the user set the rotor position, it acted as the key for encrypting the data, and the correct cipher text would decode only if the receiver had the correct rotor position.

When the user set the rotor position, it changed the substitution array for that particular rotor. For example, if the right rotor position was set to 'B', then the substitution array would start from 'K' and end with 'D'. The character 'B' would point towards the start of the substitution array, and the characters on the left side of 'B' would point to the end of the remaining characters. This same principal would follow in all the substitutions of all the rotors. The substitution tables for the right, middle, and left rotors were different, and each rotor would perform a substitution based on its respective substitution table.

Initially the User has to set the rotors position to encrypt the data. For example, if the user has select the B as the right rotor position, then the right rotor substitution will change like this:



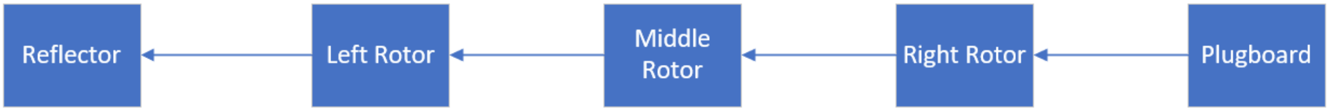
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

K M T F O I X W V U B A Q P N Y E C R J S L Z H G D

So the D will come to the first position and the K and M will come to the second and third position and all the substitute letters will go one position forward. This same technique will follow for the change in substitution of all rotors.

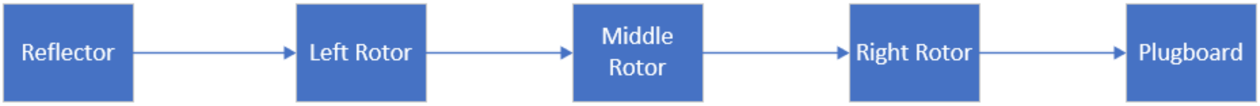
The Flow of operations when the substitution is happening:

The flow will substitute char with respect to their number in the substitution array



The flow of operations when the substituted char is replaced with the alphabetic character:

The flow will substitute the alternate alphabet in place of the substituted char



Reflector:

The reflector is the final part of the Enigma machine, and it performs the same function as the rotors. It gives the substitution of the character at the index. However, the reflector always had the same substitution table, and it was the same for all Enigma machines. It was the only part of the machine that remained static, which made it easier for cryptanalysts to crack the code. The reflector would send the encrypted letter back through the rotors and plugboard, where the substitution process would be repeated, resulting in the final encrypted letter.

The substitution in place of alphabets in the rotor:

Alphabet: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Substitution: Y R U H Q S L D P X N G O K M I E B F Z C W V J A T

Implementation Notes

The machine has the unique encryption method so each part was a challenge to implement.

i. Size of the User Entered Input Plain Text.

When the user is entering plain text for encryption so don’t have idea what will be the size of that. Every time plain text will be going to be of different size and in the documentation it was clearly mentioned that minimum size of the user given input should be 600 characters.

After much surfing and brainstorming the solution for the problem came to my mind.

Solution: Dynamic Memory Allocation

The Dynamic memory will allocate the memory on the run time according to the user requirement.

ii. Setting Rotors Position according to the user given input:

According to the document of assignment, user must be able to set the initial position of the rotors. This will change the rotors substitution and user will be able to use it as the encryption key. The cipher text will only return to its plain text if the user gives the correct position of the rotors.

Solution: Changing the substitution array values to move one position forward

iii. Encrypting and decrypting space character:

This was challenge because when decrypting the text if we are unable to decrypt the text with the right space at the time, the user will not identify the plain text.

Solution: Using the . character at the space when encrypting the text

So when decrypting the text, I told the program to substitute the ‘ ’ in place of ‘.’

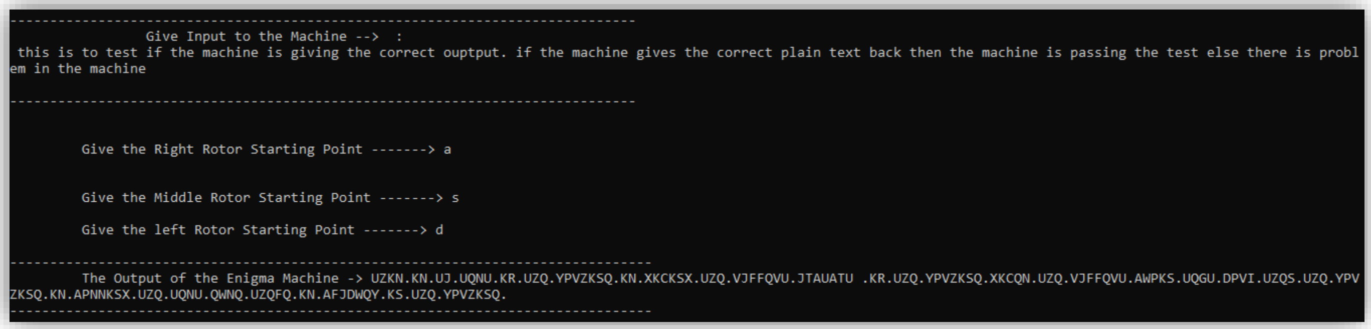
Testing of the Machine

Following test of the machine were carried out

i. Giving Different size plain text to validate the output of the machine

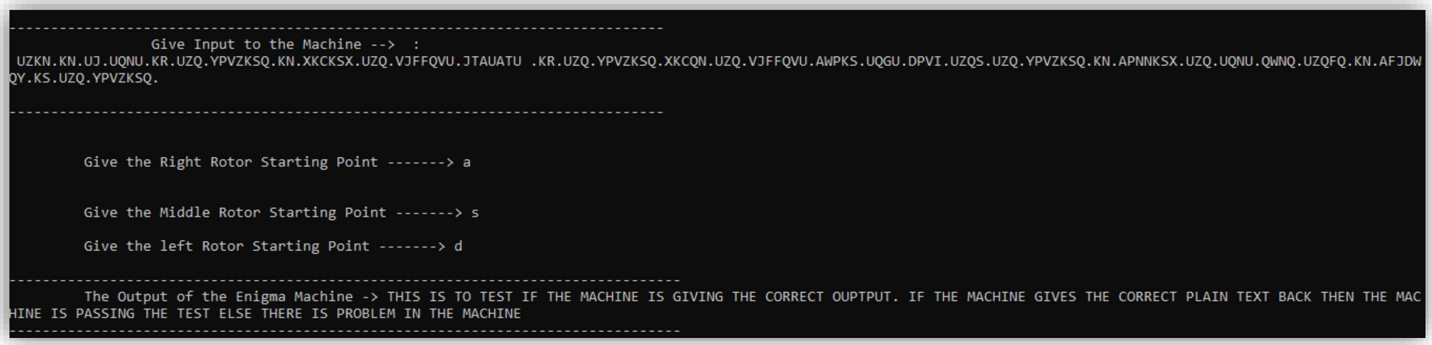
This test was carried out to check if the machine is able to read all the given input plain text to encrypt it all and then giving the cipher of the plain text to validate if the machine decrypts the same plain text. Initially, it was giving wrong answers because of the wrong algorithm. After some work I was able to correct out all the mistakes and then finally the machine was giving the correct answers.

This is encrypting the text



```
------------------------------------------------------------------------------
                 Give Input to the Machine -->  :
 this is to test if the machine is giving the correct ouptput. if the machine gives the correct plain text back then the machine is passing the test else there is probl
em in the machine

------------------------------------------------------------------------------


         Give the Right Rotor Starting Point -------> a


         Give the Middle Rotor Starting Point -------> s

         Give the left Rotor Starting Point -------> d

--------------------------------------------------------------------------------
         The Output of the Enigma Machine -> UZKN.KN.UJ.UQNU.KR.UZQ.YPVZKSQ.KN.XKCKSX.UZQ.VJFFQVU.JTAUATU .KR.UZQ.YPVZKSQ.XKCQN.UZQ.VJFFQVU.AWPKS.UQGU.DPVI.UZQS.UZQ.YPV
ZKSQ.KN.APNNKSX.UZQ.UQNU.QWNQ.UZQFQ.KN.AFJDWQY.KS.UZQ.YPVZKSQ.
--------------------------------------------------------------------------------
```

This is decrypting the text.



```
------------------------------------------------------------------------------
                 Give Input to the Machine -->  :
 UZKN.KN.UJ.UQNU.KR.UZQ.YPVZKSQ.KN.XKCKSX.UZQ.VJFFQVU.JTAUATU .KR.UZQ.YPVZKSQ.XKCQN.UZQ.VJFFQVU.AWPKS.UQGU.DPVI.UZQS.UZQ.YPVZKSQ.KN.APNNKSX.UZQ.UQNU.QWNQ.UZQFQ.KN.AFJDW
QY.KS.UZQ.YPVZKSQ.

------------------------------------------------------------------------------


         Give the Right Rotor Starting Point -------> a


         Give the Middle Rotor Starting Point -------> s

         Give the left Rotor Starting Point -------> d

--------------------------------------------------------------------------------
         The Output of the Enigma Machine -> THIS IS TO TEST IF THE MACHINE IS GIVING THE CORRECT OUPTPUT. IF THE MACHINE GIVES THE CORRECT PLAIN TEXT BACK THEN THE MAC
HINE IS PASSING THE TEST ELSE THERE IS PROBLEM IN THE MACHINE
--------------------------------------------------------------------------------
```

The plain text and the decryption text are same so the machine is passing the test of reading all the plain text characters and then decrypting them correctly.

Reflective Analysis

This project gives me the confidence and strong grip on the c language concepts and helps me a lot to built logic for a problem.

Initially, it was hard to built logic and develop the flow of the control of the operations of the parts of the machine, but after spending a couple day I was successful to develop logic or the flow of the C++ functions.

In the start, choosing the right substitution array for the plugboard, rotors and the reflector was a point of great concern and I surf the websites to find the appropriate substitution lists. All the substitution lists for all the parts are picked from the internet, while rest of the program is coded by myself. When I selected the right substitution arrays then I draw them on the paper to check whether this substitution is providing the correct encryption and decryption of the user given input. After being satisfied by the correct output of the substitution combination of different parts, then I started coding them in c.

The last and most difficult part to implement was the back substitution from the reflector to the plugboard. This part takes more couple of days to build the logic and the algorithm to select the right technique for the encryption and decryption. However, I was able to solve it at last.

The success of this project has brought more ideas to my mind to build more cryptographic tools as these are the source to build strong logic and strong grip on the programming languages. Next time I am making my mind to build a tool which uses the asymmetric encryption where there are two keys different keys for encryption and decryption.