Nile University

ECEN-502 Introduction to Computer Security Project Report

Classical Encryption Techniques Implementation and Comparison

Under the Supervision of

Dr. Tawfik Ismail Tawfik

Eng. Aya Mohamed Safan

By

Adham Ahmed Abdallah	**18100846**
Mohamed Salah	**18101904**
Mohammad Gamal	**18100693**
Mohammad Tariq	**18102044**
Alaa Ibrahim	**18101242**

Submitted in partial fulfillment of the requirements

Table of Contents

Table of figures

I. Executive Summary

In this paper, the implementation and comparison between classical encryption techniques is discussed. First, a brief introduction on encryption techniques, how they work and how they started. Then, methods of encrypting and encrypting texts using these techniques and comparison between them in terms of how hard to decipher texts. In the Methodology, the techniques are implemented upon programming and mathematical levels. For this purpose, a C++ program is written to implement these techniques and outputting the text after ciphering/deciphering it.

II. Background Information

A. Monoalphabetic cipher

A monoalphabetic cipher is a simple substitution cipher in which each letter of the plaintext is replaced with a fixed corresponding letter of the ciphertext. The key for a monoalphabetic cipher is a one-to-one mapping between the plaintext letters and the ciphertext letters. For example, consider the following monoalphabetic cipher:

Plaintext: abcdefghijklmnopqrstuvwxyz Ciphertext: qwertyuiopasdfghjklzxcvbnm

In this cipher, the letter "a" is replaced with "q", "b" is replaced with "w", "c" is replaced with "e", and so on. To encode a message using this cipher, one would simply substitute each letter of the plaintext with its corresponding letter in the ciphertext.

Monoalphabetic ciphers are relatively easy to break because they do not change the frequency of occurrence of the letters in the plaintext. This means that an attacker can use statistical analysis to determine the most likely mapping between the plaintext and ciphertext letters and use that mapping to decrypt the message. Despite their relative simplicity, monoalphabetic ciphers have been used for centuries as a means of secure communication. They are still used today in some simple codes and puzzles, but they are generally not considered secure for real-world applications.

B. Caesar cipher

The Caesar cipher, also known as the shift cipher, is a monoalphabetic substitution cipher in which each letter of the plaintext is shifted a fixed number of places down the alphabet. The Caesar cipher is named after Julius Caesar, who is said to have used it to communicate with his officers. To encode a message using the Caesar cipher, one would choose a key (a positive integer) and then shift each letter of the plaintext by that number of places down the alphabet. For example, if the key is 3 and the plaintext is "hello", the ciphertext would be "khoor". To decode the message, one would simply shift the ciphertext letters back by the same number of places. Here is an example of the Caesar cipher with a key of 3:

Plaintext: abcdefghijklmnopqrstuvwxyz Ciphertext: defghijklmnopqrstuvwxyzabc

The Caesar cipher is relatively easy to break because it does not change the frequency of occurrence of the letters in the plaintext. An attacker can use statistical analysis to determine the most likely key and use that key to decrypt the message. Despite its simplicity, the Caesar cipher was widely used in the past and is still used today in some simple codes and puzzles. However, it is not considered secure for real-world applications.

C. Vigenère cipher

The Vigenère cipher is a polyalphabetic substitution cipher that was invented in the 16th century by Blaise de Vigenère. It is a more secure variant of the Caesar cipher, which is a monoalphabetic substitution cipher. In the Vigenère cipher, a secret keyword is used to create a repeating key that is used to encrypt the plaintext. The key consists of a sequence of letters, and each letter in the key corresponds to a particular shift in the alphabet. To encode a message using the Vigenère cipher, one would first choose a secret keyword and repeat it until it is as long as the plaintext. For example, if the keyword is "SECRET" and the plaintext is "HELLO", the key would be "SECRE". Then, each letter of the plaintext is shifted by the corresponding number of places indicated by the corresponding letter in the key. For example, if the key is "SECRE" and the plaintext is "HELLO",

the ciphertext would be " ZINCS ". Here is an example of the Vigenère cipher with a key of "SECRET": Plaintext: HELLO Key: SECRE Ciphertext: ZINCS

The Vigenère cipher is more secure than the Caesar cipher because it uses a different shift for each letter of the plaintext, making it more difficult to break using statistical analysis. However, it is still vulnerable to certain attacks, such as the Kasiski examination, which can be used to determine the length of the key and subsequently break the cipher. Despite its improved security, the Vigenère cipher is no longer considered secure for real-world applications, as there are more secure ciphers available. However, it is still of historical interest and is occasionally used in simple codes and puzzles.

D. Row transposition

A row transposition cipher is a simple transposition cipher in which the plaintext is written in a grid, and the rows of the grid are then rearranged to create the ciphertext. The key for a row transposition cipher is the order in which the rows are rearranged. To encode a message using a row transposition cipher, one would first write the plaintext in a grid, with a fixed number of letters per row. The number of rows and columns in the grid is determined by the length of the plaintext and the size of the key. For example, if we choose a key of "3142", we can write the plaintext in a 4x6 grid. To create the ciphertext, we rearrange the rows of the grid according to the key. In this case, the key "3142" means that the third row should be moved to the first position, the first row should be moved to the second position, the fourth row should be moved to the third position, and the second row should be moved to the fourth position. To decode the message, one would simply rearrange the rows of the ciphertext according to the inverse of the key. In this case, the inverse of the key "3142" is "2413", which means that the second row should be moved to the first position, the fourth row should be moved to the second position, the first row should be moved to the third position, and the third row should be moved to the fourth position. Row transposition ciphers are relatively simple to implement and can be quite effective if the key is kept secret. However, they are vulnerable to attacks such as the Railway Track Attack, which can be used to determine the key and decrypt the message.

E. Playfair matrix

The Playfair cipher is a polyalphabetic substitution cipher that was invented in the 19th century by Charles Wheatstone. It is based on the idea of using a 5x5 matrix, called the Playfair square, to encode pairs of letters. To encode a message using the Playfair cipher, one would first create a 5x5 matrix using a secret keyword. The matrix is filled in with the letters of the alphabet, starting with the keyword and then filling in the remaining letters in alphabetical order. For example, if the keyword is "SECRET", the matrix would be:

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

To encode a message, the plaintext is first broken into pairs of letters (if the plaintext has an odd number of letters, an "X" is added to the end to make it a multiple of two). For each pair of letters, the following rules are applied:

- If the letters are the same, an "X" is inserted between them. For example, "EE" becomes "EX".
- If the letters are in the same row of the matrix, the letters in the same columns are used to create the ciphertext.
- If the letters are in the same column of the matrix, the letters in the same rows are used to create the ciphertext.
- If the letters are in different rows and columns of the matrix, the letters in the same rows and the same columns as the original letters are used to create the ciphertext.

To decode a message, the same rules are applied in reverse.

The Playfair cipher is more secure than simple substitution ciphers because it encodes pairs of letters instead of individual letters. However, it is still vulnerable to certain attacks, such as frequency analysis, which can be used to determine the most likely plaintext letters and use that information to decrypt the message. Despite its improved security, the Playfair cipher is no longer considered secure for real-world applications, as there are more secure ciphers available. However, it is still of historical interest and is occasionally used in simple codes and puzzles.

III. Methodology

A. Main program:

The console application is implemented as follows using the below flow chart diagram:

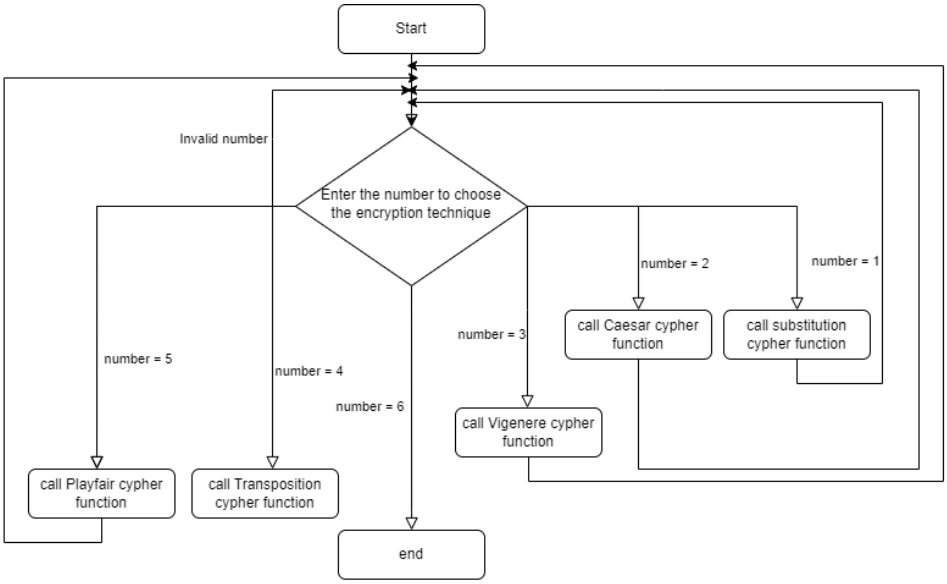

Figure 1 Console application flowchart

When the console application starts, the user is prompted to enter a number from 1 to 6 as shown in the figure below. The first 5 numbers correspond to the 5 encryption techniques we use: Substitution Cipher, Caesar Cipher, Vigenère Cipher, Row transposition cipher, Playfair cipher. The sixth number is when the user wants to terminate the program. If the user enters a different number from these, the screen prints error message as follows "Wrong choice! Try again "and prompts the user to try again. Every technique has its own function, and the function is called directly after the user enters the correct number.

```
Enter the number of the encryption technique you want to use
1. Substitution Cypher
2. Caesar Cypher
3. Vigenere Cypher
4. Transposition Cypher
5. Playfair Cypher
6. End the program
7
Wrong choice! Try again
Enter the number of the encryption technique you want to use
1. Substitution Cypher
2. Caesar Cypher
3. Vigenere Cypher
4. Transposition Cypher
5. Playfair Cypher
6. End the program
```

Figure 2 snippet of the program output

B. Substitution Cipher

The cipher function is implemented as follows using the below flow chart diagram:

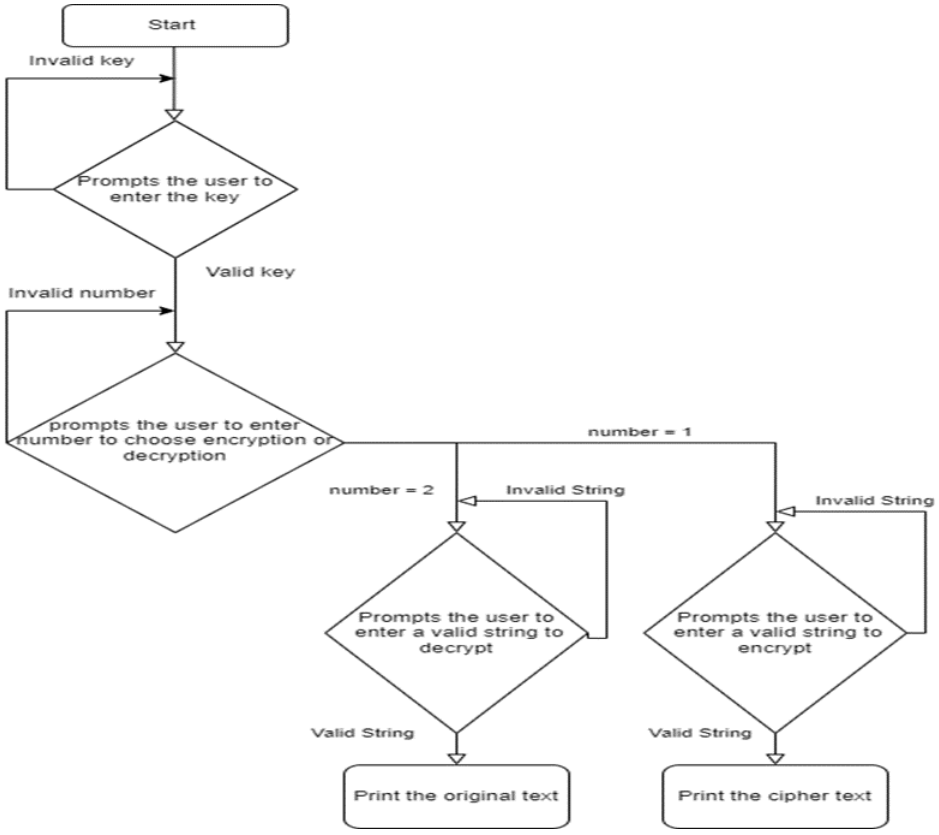

Figure 3 Substitution cipher flowchart

When the substitution cipher is called, the user is prompted first to enter a valid key which consist of 26 keys corresponding to every letter, if the key is invalid the user is asked to try again. Then the user is asked to enter a number from 1 or 2 to choose whether to encrypt or decrypt the text as shown in the figure below. If the user enters an invalid number, it is asked to enter a valid number. If the number is one the program replaces each letter with its corresponding in the key entered by the user, else it replaces each letter with its corresponding letter in the text based on the key given from the user. Then the final text is printed.

C. Caesar Cipher

The cipher function is implemented as follows using the below flow chart diagram:

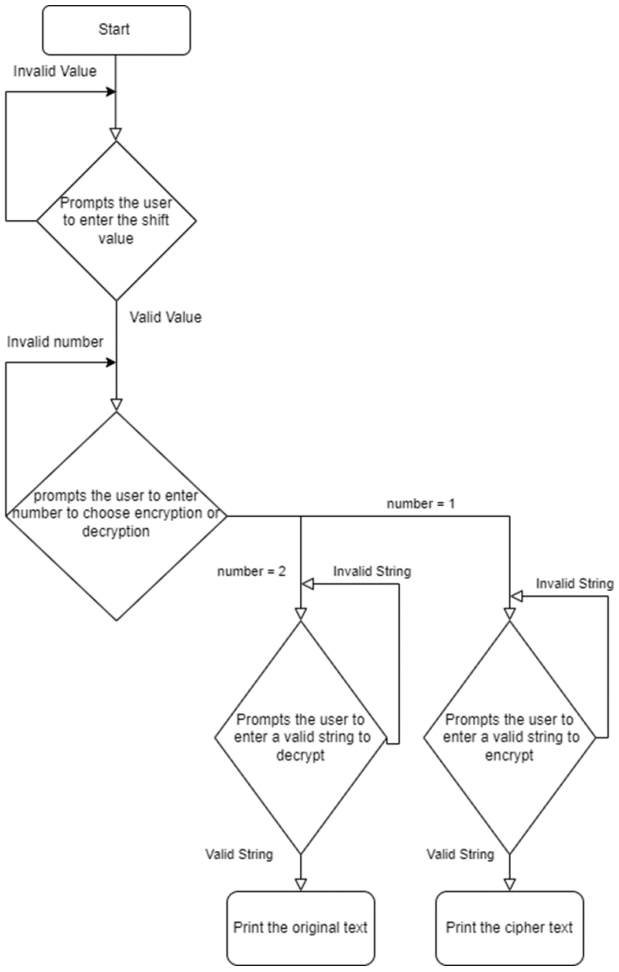

Figure 4 Caesar cipher flowchart

When the Caesar cipher is called, the user is prompted first to enter a valid shift value which consists of a single number. If the shift value is not a number, the user is asked to try again. Then the user is asked to enter a number from 1 or 2 to choose whether to encrypt or decrypt the text as shown in the figure below. If the user enters an invalid number, it is asked to enter a valid number. If the number is one the program adds the shift value to every letter in the text and take the resulting number modulo 26, else it subtracts from each letter the shift value and take the resulting number modulo 26. Then the final text is printed.

D. Vigenère Cipher

The cipher function is implemented as follows using the below flow chart diagram:

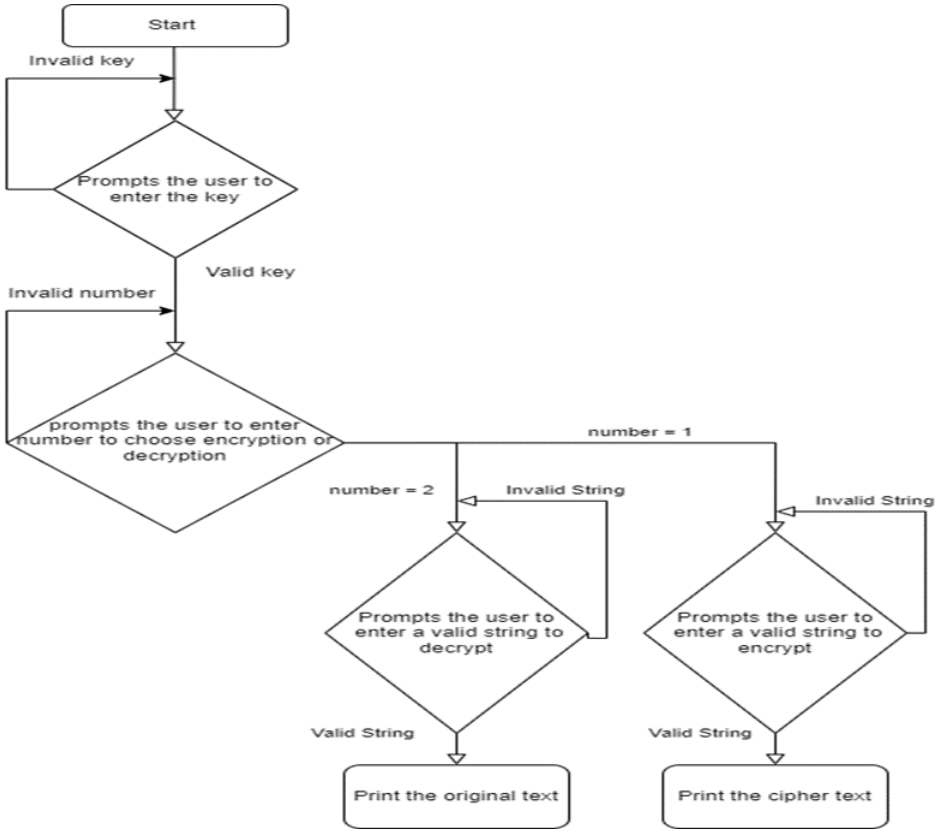

Figure 5 Vigenère cipher flowchart

When the Vigenère cipher function is called, the user is prompted first to enter a valid key which consist of letters only, if the key is invalid the user is asked to try again. Then the user is asked to enter a number from 1 or 2 to choose whether to encrypt or decrypt the text as shown in the figure below. If the user enters an invalid number, it is asked to enter a valid number. After the number is entered the key must be extended to be the same size as the text given from the user. If the number is one the program replaces each letter with its addition with the corresponding key letter and take the result modulo 26. Otherwise, it replaces each letter with its value subtracted by the corresponding key letter and take the result modulo 26. Then the final text is printed.

E. Transposition Cipher

The cipher function is implemented as follows using the below flow chart diagram:

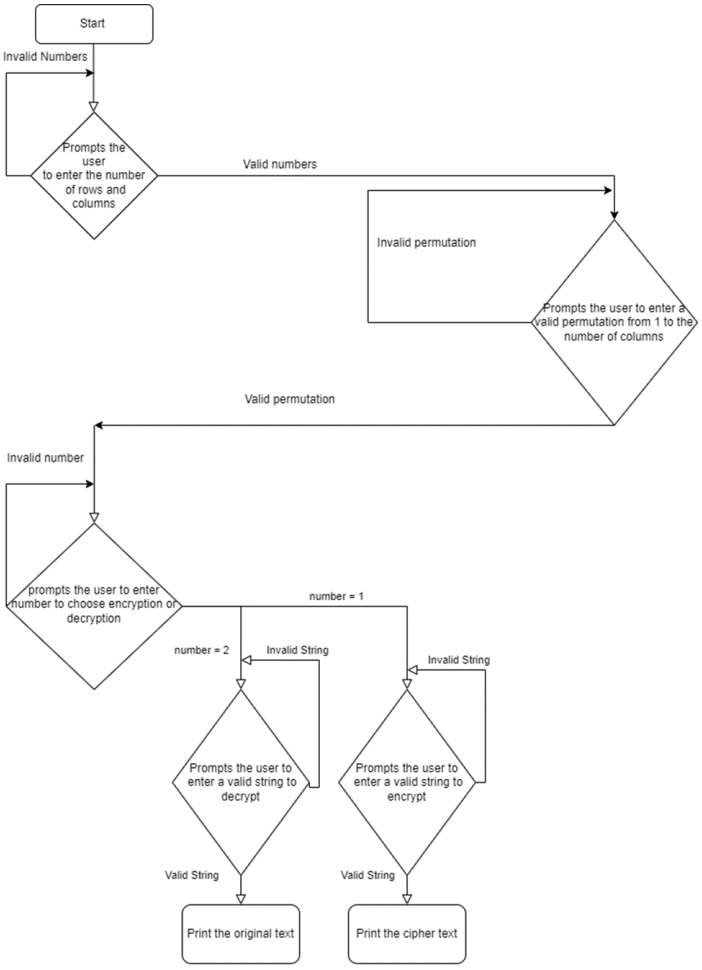

Figure 6 Transposition cipher flowchart

When the row transposition cipher function is called, the user is prompted first to enter a valid number of rows and columns which represents the matrix of the text, if the input numbers are invalid the user is asked to try again. Then the user is asked to enter a valid permutation of the columns, if the permutation is invalid the user is asked to try again. Then the user is asked to enter a number from 1 or 2 to choose whether to encrypt or decrypt the text. If the user enters an invalid number, it is asked to enter a valid number. After the number is entered the matrix of the text initialized based on the number of rows and columns. If the number is one the program permutes the columns according to the given permutation and then assembles the text from the matrix. Otherwise, the program permutes the columns according to the inverse of the given permutation and then assembles the text from the matrix. Then the final text is printed.

F. Playfair cipher

The cipher function is implemented as follows using the below flow chart diagram:

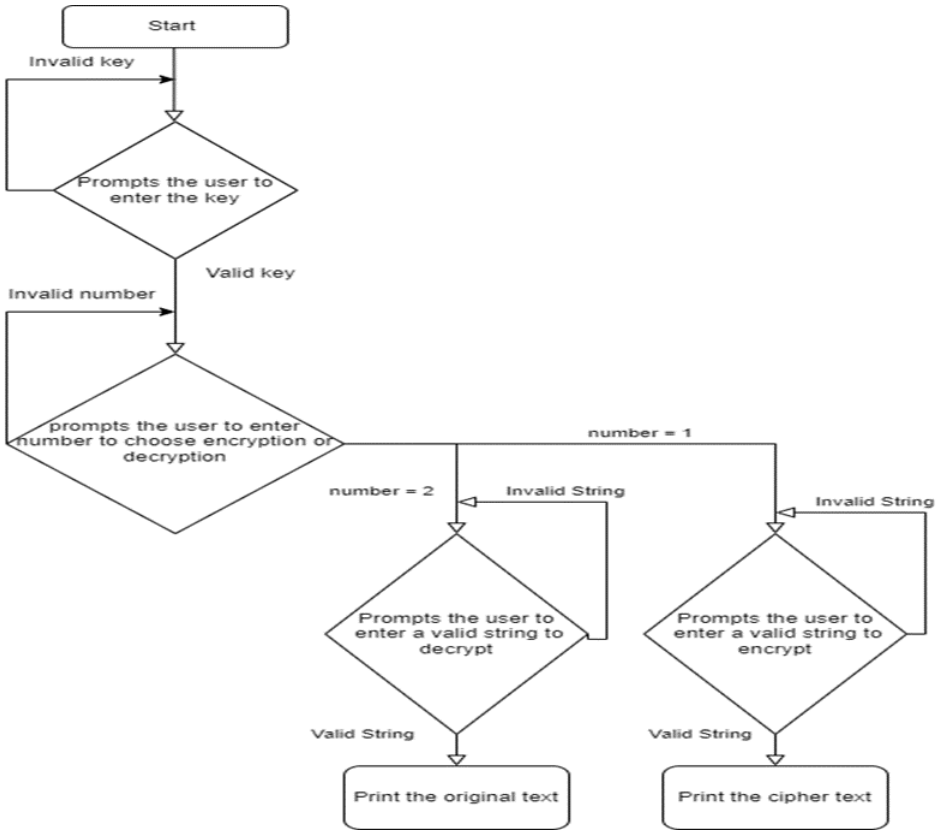

Figure 7 Playfair cipher flowchart

When the playfair cipher function is called, the user is prompted first to enter a valid key which consists of letters only, if the key is invalid the user is asked to try again. Then the user is asked to enter a number from 1 or 2 to choose whether to encrypt or decrypt the text. If the user enters an invalid number, it is asked to enter a valid number. After the number is entered the 5x5 matrix of letters is initialized based on the key and each letter has its own position in the matrix. Then the text is split into consecutive pairs. If the number is one the program replaces each pair of letters with their corresponding letters in the matrix. Otherwise, it replaces each pair of consecutive letters with the reverse operation that happened in number 1. Then the final text is printed.

II. Results and Discussions

We have tried many test cases to test whether the program is correct or not. We will show a sample test case for each method.

Substitution Cipher

```
1
Enter they key ( 26 letters corresponding for each letter )
DKV QFIBJWPES CXHT MY AU OLR GZN
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
1
Enter the string you want to encrypt
IF WE WISH TO REPLACE LETTERS
WI RF RWAJ UH YFTSDVF SFUUFYA
```

Figure 8 substitution cipher encryption

```
1
Enter they key ( 26 letters corresponding for each letter )
DKV QFIBJWPES CXHT MY AU OLR GZN
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
2
Enter the string you want to decrypt
WI RF RWAJ UH YFTSDVF SFUUFYA
IF WE WISH TO REPLACE LETTERS
```

Figure 9 substitution cipher decryption

Caesar *Cipher*

```
2
Enter the shift value
3
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
1
Enter the string you want to encrypt
meet me after the toga party
phhw ph diwhu wkh wrjd sduwb
```

Figure 10 Caesar cipher encryption

```
2
Enter the shift value
3
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
2
Enter the string you want to decrypt
phhw ph diwhu wkh wrjd sduwb
meet me after the toga party
```

Figure 11 Caesar cipher decryption

Vigenère Cipher

```
3
Enter the key
LEMON
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
1
Enter the string you want to encrypt
ATTACK AT DAWN
LXFOPV EF RNHR
```

Figure 12 Vigenère cipher encryption

```
3
Enter the key
LEMON
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
2
Enter the string you want to decrypt
LXFOPV EF RNHR
ATTACK AT DAWN
```

Figure 13 Vigenère cipher decryption

Row Transposition

```
4
Enter the rows and columns of the matrix
4 7
Enter a valid permutation of the columns
4 3 1 2 5 6 7
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
1
Enter the string you want to encrypt
attack postponed until two am
ttna aptm tsuo aodw coi* knl* pet*
```

Figure 14 Row transposition encryption

```
4
Enter the rows and columns of the matrix
4 7
Enter a valid permutation of the columns
4 3 1 2 5 6 7
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
2
Enter the string you want to decrypt ( enter * for empty characters )
ttna aptm tsuo aodw coi* knl* pet*
a t t a c k p o s t p o n e d u n t i l t w o a m * * *
```

Figure 15 Row transposition decryption

Playfair Cipher

```
5
Enter the key
playfair example
Choose the number whether you want to encrypt or decrypt.
1. Encrypt
2. Decrypt
1
Enter the string you want to encrypt
Hide the gold in the tree stump
Bm od zb xd na be ku dm ui xm mo uv if
```

Figure 16 Playfair cipher encryption

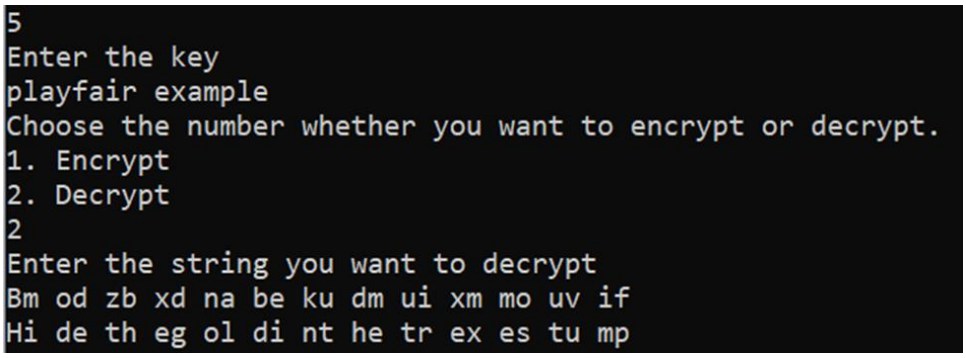

Figure 17 Playfair cipher decryption

IV. Conclusion and Future work

We have successfully implemented and tested 5 different classical encryption techniques and compared between them considering the way of hacking the text without knowing the key of ciphering. We have tried multiple test cases and they are all working and producing the correct output. We hope in the future to implement more classical techniques and build a GUI application instead of the console application we have made.