King Abdulaziz University
Faculty of Engineering
Electrical and Computer Engineering Department

Discrete Math. Project Report

Keyword Columnar Transposition

(EE305)

Team Number: 4

Name	ID
Mohannad Adel Alnahhas	1741538
Khaled Waleed Saqi	1741869
Mohammed Abdullah Alsaggaf	1740489

Fall 2019 Instructor: Dr.Emad Khalaf

1. Introduction

Data transmission can be interrupted by a third person who may manipulate it. This is dangerous in the perspective of important data or massages. To protect data from manipulating, we use some of encrypting methods. One of the methods is keyword columnar transposition. The idea of this method is to have a keyword and a sentence, where the sentence will be arranged with the keyword. Resulting in an encoded message. The aim of this project is to construct an application in which the input is the keyword and the sentence while the output is either the encoded or decoded message. Finally, what you will find in this report is the methodology of encoding and decoding the message, programming algorithm, and conclusion.

2. Mathematical Methodology

In the begging there are two important things which are keyword and Sentence that we want to encode or decode. First to encode a sentence, the keyword and the sentence must be inserted in an array. The number of columns equal to the number of the keyword letters Eq (1), and the number of rows equal to the number of the sentence letters divided by the keyword letters number Eq (2).

$$Columns = Keyword\ Letters \text{ ---------Eq (1)}$$

$$Rows = \frac{Sentence\ Letters}{Keyword\ Letters} \text{ ----------Eq (2)}$$

After that, the sentence will be inserted in array row by row. any empty cells will be inserted with "X". Now, the last step is arranging the keyword regarding to alphabet order. The encrypt message can be read column by column.

To decode a message, the message must be inserted column by column in an array. The number columns and rows are using Eq(1) and Eq(2).Then, rearrange the array as the same order for the keyword. The decoded message can be read row by row.

3. Programming Algorithm

Programming algorithm is the core of any program, the more optimized algorithm operates the program with less time, less memory usage and more accurate results. However, the integrated development environment (IDE) does not make any

difference as long as correct results is shown. In this program the Java Development Kit (JDK) was used to develop this algorithm and NetBeans as IDE. NetBeans has been chosen in order to have an easy and well-designed graphical user interface (GUI). The algorithm used in this program is based on four main parts:

a) Initialize Arrays for encryption and decryption.
b) Encrypt the message array.
c) Decrypt the message array.
d) Graphical User Interface.

3.a. Initialize Arrays

For easier encryption and decryption, the program generates four arrays as follows: **Keywordarray**, **Sentancearray**, **Unsortedarray** and **Sortedarray**. As shown in figure 1 & figure 2, **Keywordarray** and **Sentancearray** will receive their data from the user and **Unsortedarray** will be generated automatically.

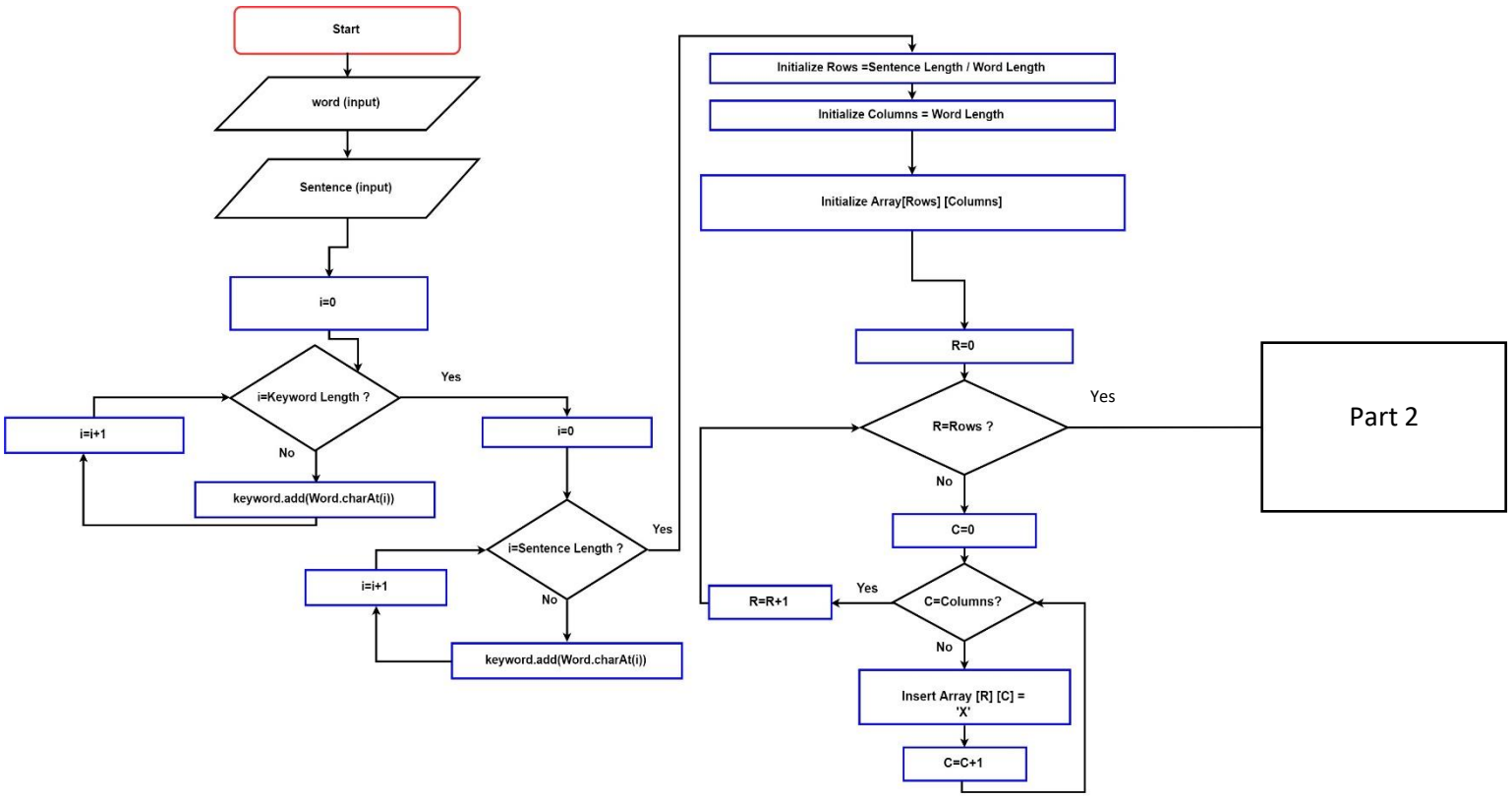

Figure 1: Initialize Arrays Flowchart (Part 1)

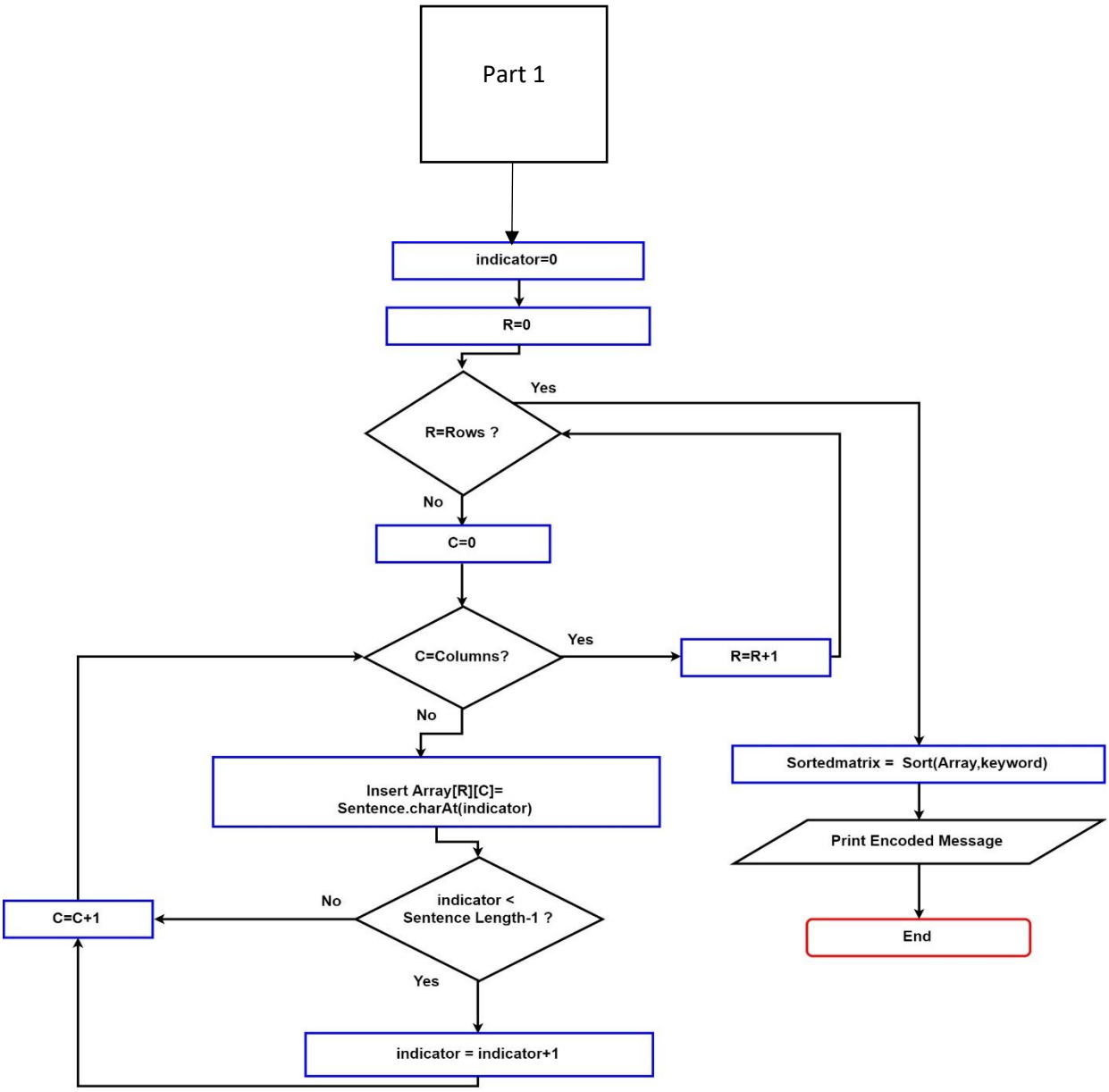

Figure 2: Initialize Arrays Flowchart (Part 2)

3.b. Encrypt the Message Array

A method called **Sort** will receive both **Keywordarray** and **Unsortedarray** to store the encoded message in the **Sortedarray**. As shown in figure 3, the **Sort** method will sort the **Keywordarray** with alphabet order. The sorted **Keywordarray** will be compared with unsorted **Keywordarray** to specify the index for each letter and then to reorder the **Unsortedarray** into the **Sortedarray** which generate the encoded message.

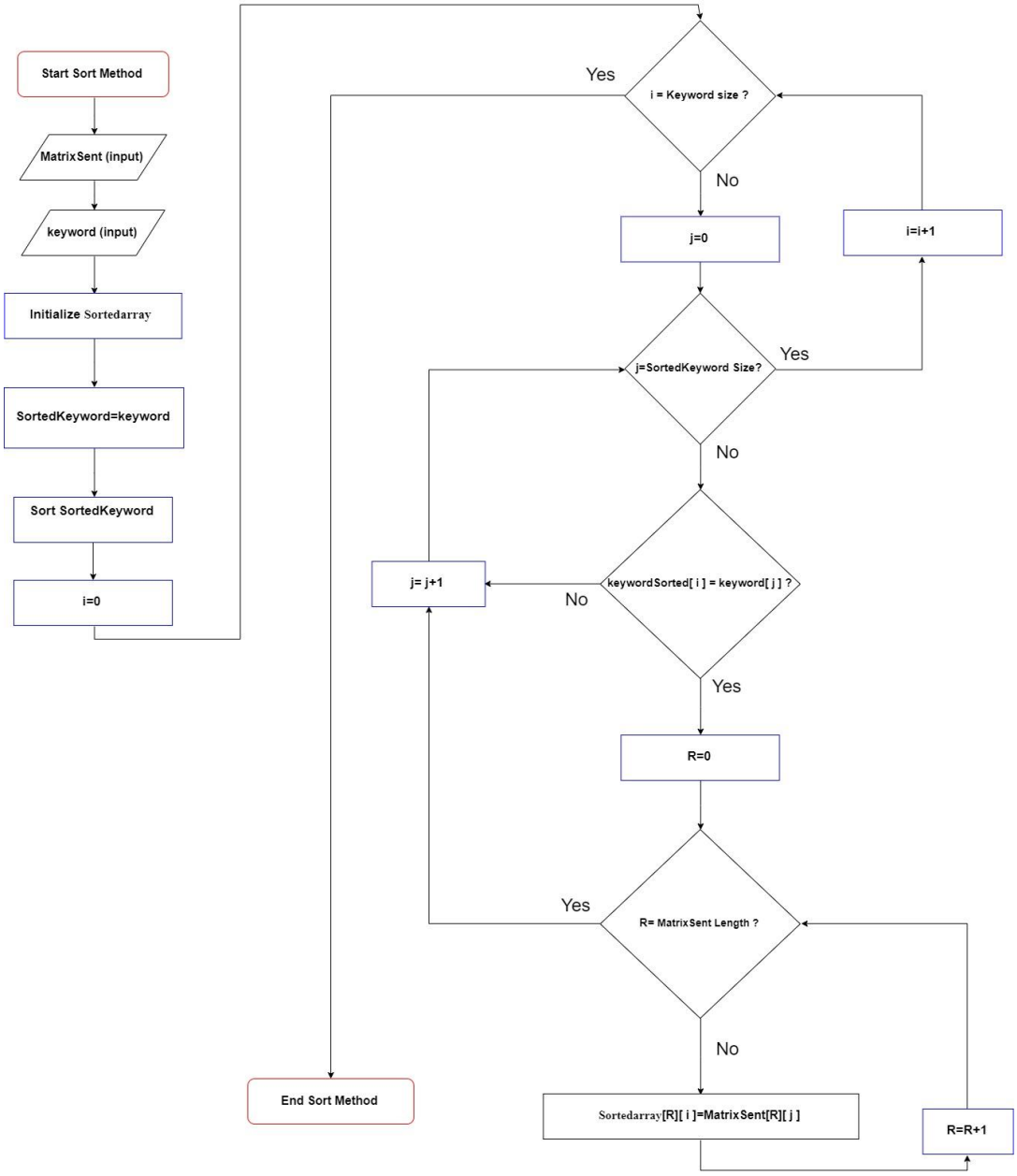

Figure 3: Encrypt the Message Array

3.c. Decrypt the Message Array

A method called **decodeSort** will receive both **Keywordarray** and **decodeM** array to store the decoded message in the **decodeSorted**. As shown in figure 4, the **decodeSort** method will sort the **Keywordarray** with alphabet order. with the same algorithm of the encoder, the sorted **Keywordarray** will be compared with unsorted **Keywordarray** to specify the index for each letter and then to reorder the **Unsortedarray** into the **Sortedarray** which generate the decoded message.

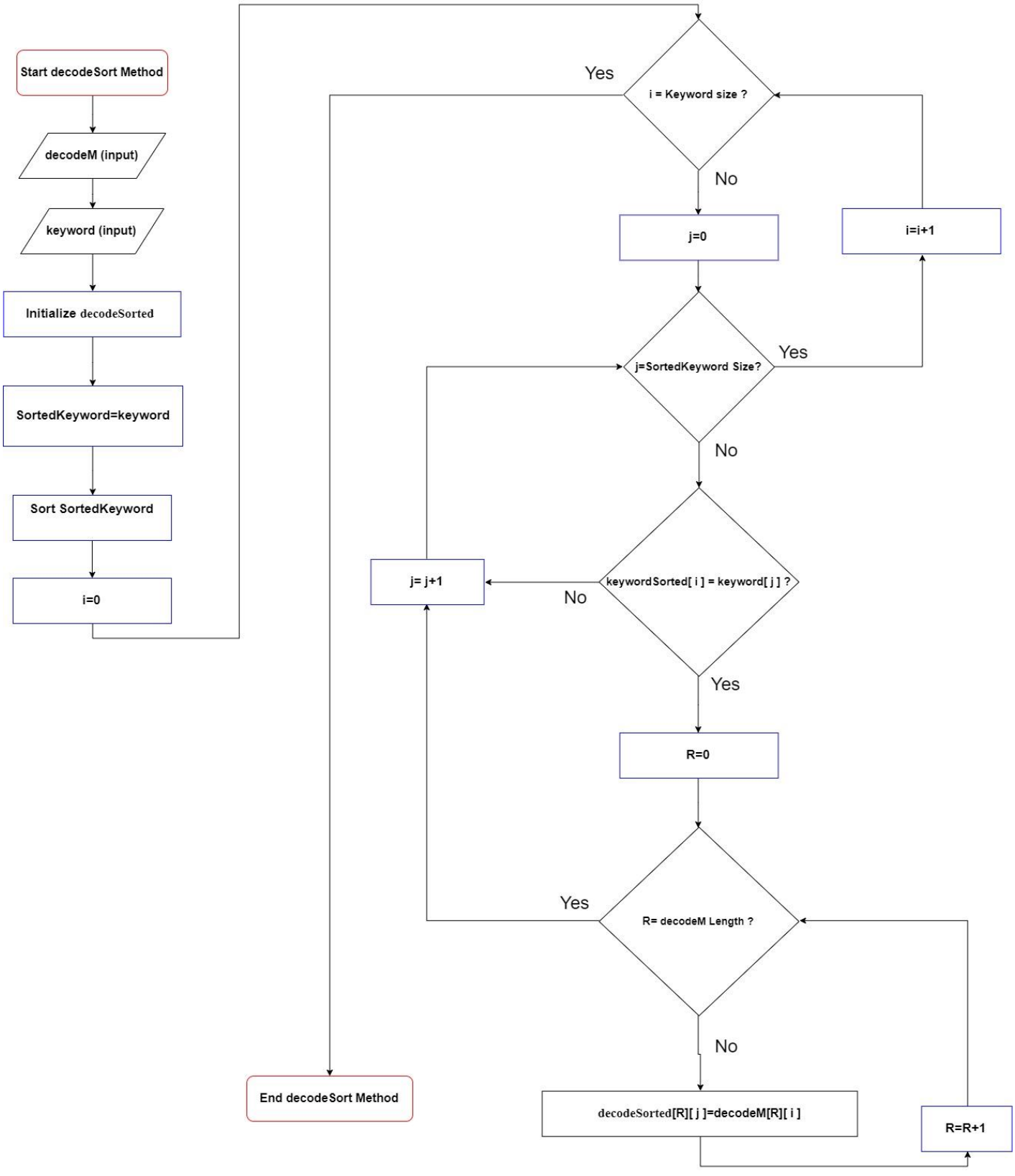

Figure 4: Decrypt the Message Array

3.d. Graphical User Interface.

To design and build the graphical user interface (GUI) NetBeans IDE was used for easier and faster design. The GUI has three frames as follow: Main Menu, Encoder Frame and Decoder Frame and all of them are fully connected which gives smooth switching between the three frames. Figure 5,6 and 7 illustrates the three designs.

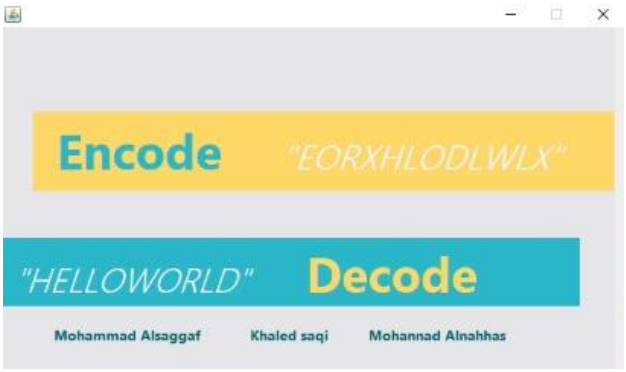

Figure 5: Main Menu Frame

Figure 6: Encoder Frame

Figure 7: Decoder Frame

4. Conclusion

By the end of the project, we became more familiar with algorithm of encoding and decoding messages using keyword columnar transposition method, and how to program it. We assured that the encoded and decoded message is correct, by solving some of the examples in the textbook. In addition, we decoded a message that was encoded by the application. We faced some problems when coding the method. One of them was how to deal with the matrix when the keyword characters doesn't divide sentence character. Which was solved by ceiling the result of sentence/keyword. Finally, the outcome of the project is how to apply theoretical knowledge we have learned in the course by coding.