



**Prifysgol Cymru**  
Y Drindod Dewi Sant  
**University of Wales**  
Trinity Saint David

**ACCA 7022**  
**ADVANCED STATISTICAL PROGRAMMING**  
**ASSIGNMENT 1**

Submitted by :Shubhi Pushkarna

Student ID: 2230035

Course: Msc Data Science and Analytics

## **TABLE OF CONTENTS**

|                                                    |    |
|----------------------------------------------------|----|
| 1. Matrix Creation.....                            | 3  |
| 2. Encrypt the message using the keyphrase.....    | 6  |
| 3. Decrypt the ciphertext using the keyphrase..... | 9  |
| 4. Questions asked from ChatGPT.....               | 12 |
| 5. References.....                                 | 13 |

## 1. MATRIX CREATION

### *Question 1.*

Create a matrix vigenere that contains the character data below. Display this matrix.

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

To create The Vigenère cipher matrix by using R studio, the R script code undergoes the following steps.

This R code uses nested loops to define the alphabet and generate a Vigenère cypher matrix. Let's dissect the code in detail:

1. **Alphabet Definition:** The first line of code defines the alphabet(alpha) as a vector with the capital letters "A" through "Z" stored in it.
2. **Constructing a Vigenère Cypher Matrix:** Next, the code uses nested loops to construct a Vigenère cypher matrix, or vigenere. It functions as follows:
  - i. The alphabet's length is allocated to n.
  - ii. The matrix is named as Vigenere and is initialised with "NA" values and the rows and columns are kept as same length of the alphabet (alpha)
  - iii. Two nested loops iterate over each row (**a**) and column (**b**) of the matrix.
  - iv. The Vigenère cypher algorithm is used to determine the value at each location [a, b] in the matrix. Following the Vigenère cypher pattern, the formula  $(a + b - 2) \% n + 1$  guarantees that the matrix is filled with the alphabetic characters.
3. **Display results:** Use print command to display the results.

### Flowchart for Task 1

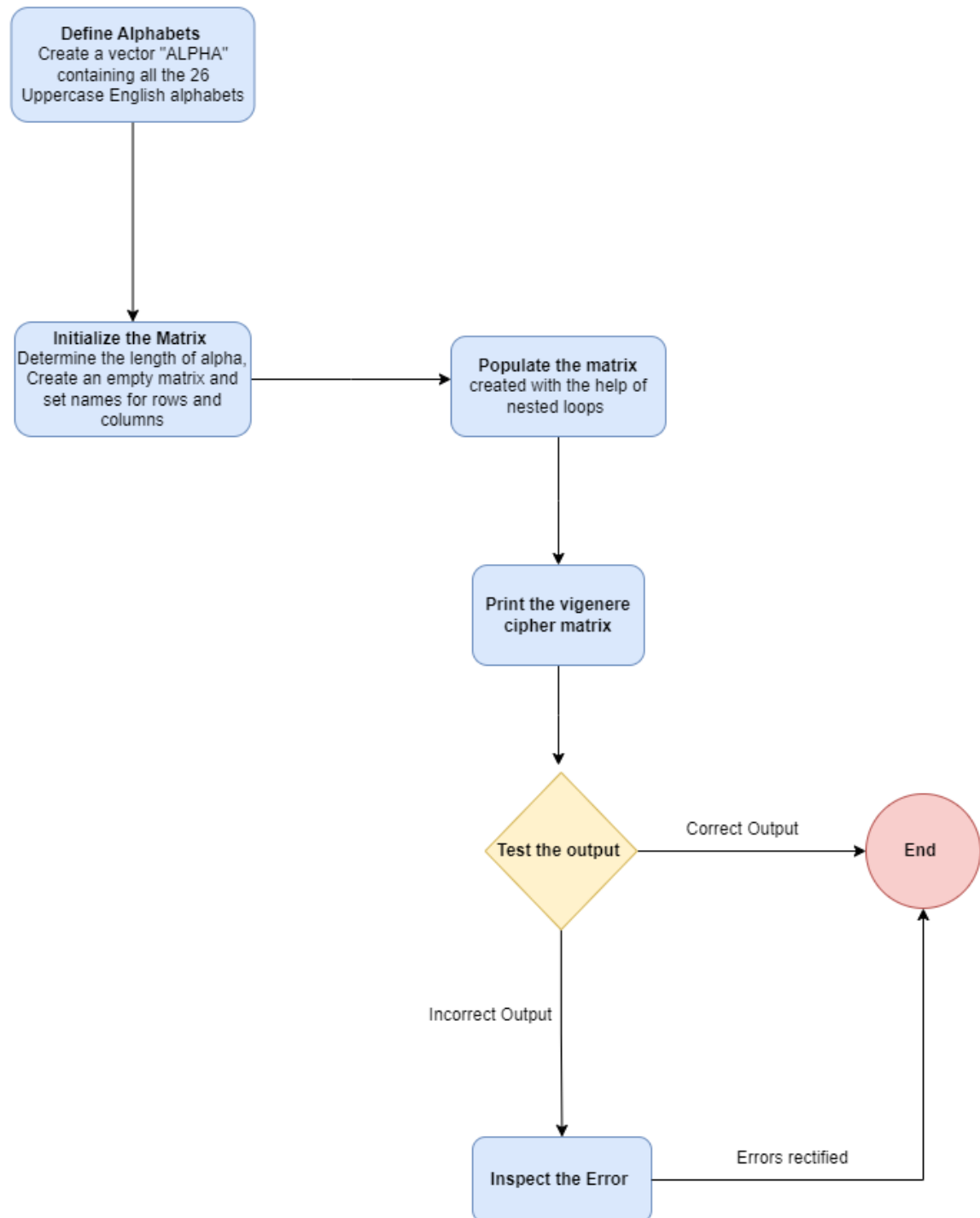


Figure 1. Flowchart for question 1

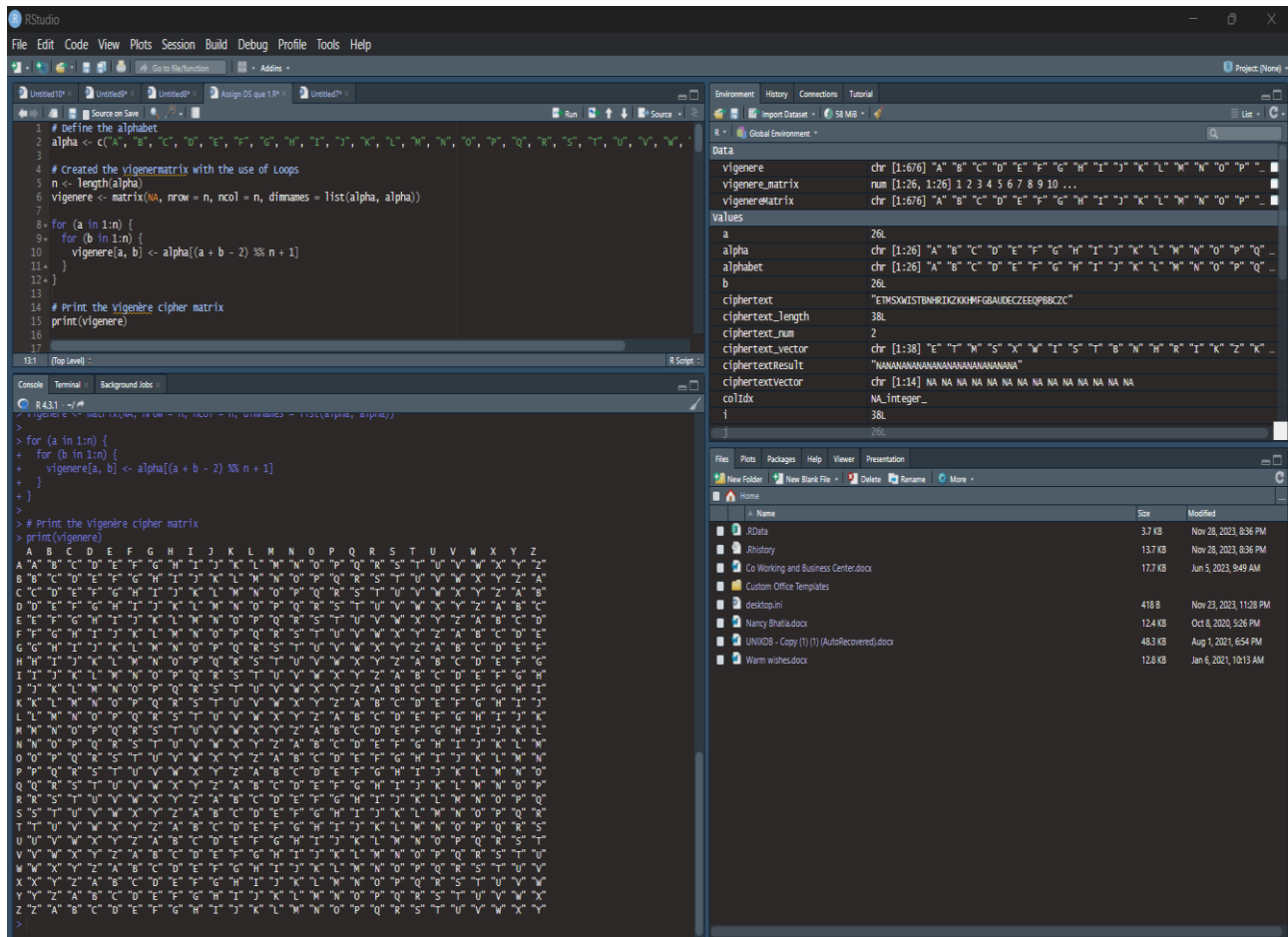


Figure 2. Screenshot of question 1

## 2. ENCRYPT THE MESSAGE USING THE KEYPHRASE

### *Question 2.*

Using the key phrase UWTSDSWANSEA, encrypt the message CAPTURETHEFLAG using the method described. You should create a vector keyphrase to hold the keyphrase, a vector plaintext to hold the plaintext and a vector ciphertext to hold the ciphertext. Display the plaintext and the ciphertext.

The following steps are undertaken for the encryption by using the keyphrase.

**Step 1. Initialising the Code:** The code defines a plaintext ("CAPTURETHEFLAG") and a keyphrase ("UWTSDSWANSEA"). The characters for the keyphrase, plaintext, and ciphertext are initialised into three vectors: keyphrase\_vector, plaintext\_vector, and ciphertext\_vector.

**Step 2. Keyphrase Repetition and Plaintext Length Matching:** The plaintext length is matched with the keyphrase repetition. A loop is used to do this, iterating through the plaintext character by character and appending the equivalent character from the keyphrase. To make sure the keyphrase is repeated cyclically, the modulo operation is employed.

**Step 3. Text to Number Conversion:** The letters in the text are translated into numerical values, with 'A' being mapped to 0, 'B' to 1, and so on. Numerical operations are carried out in this way for encryption.

**Step 4. Encryption:** The main encryption loop adds the keyphrase character's associated numeric value after converting each plaintext character to its numeric representation. Modular arithmetic is used to modify the outcome such that it is between 0 and 25. After being encrypted, the numeric value is transformed back into a character and appended to the ciphertext vector.

**Steps 5. Display Results:** The encoded ciphertext, keyphrase, and the original plaintext are all shown along with the findings at the end.

In conclusion, this code illustrates how to encrypt plaintext with a Vigenère cypher and UWTSDSWANSEA as keyphrase. The original plaintext and keyphrase are shown with the generated ciphertext.

## Flowchart for Task 2

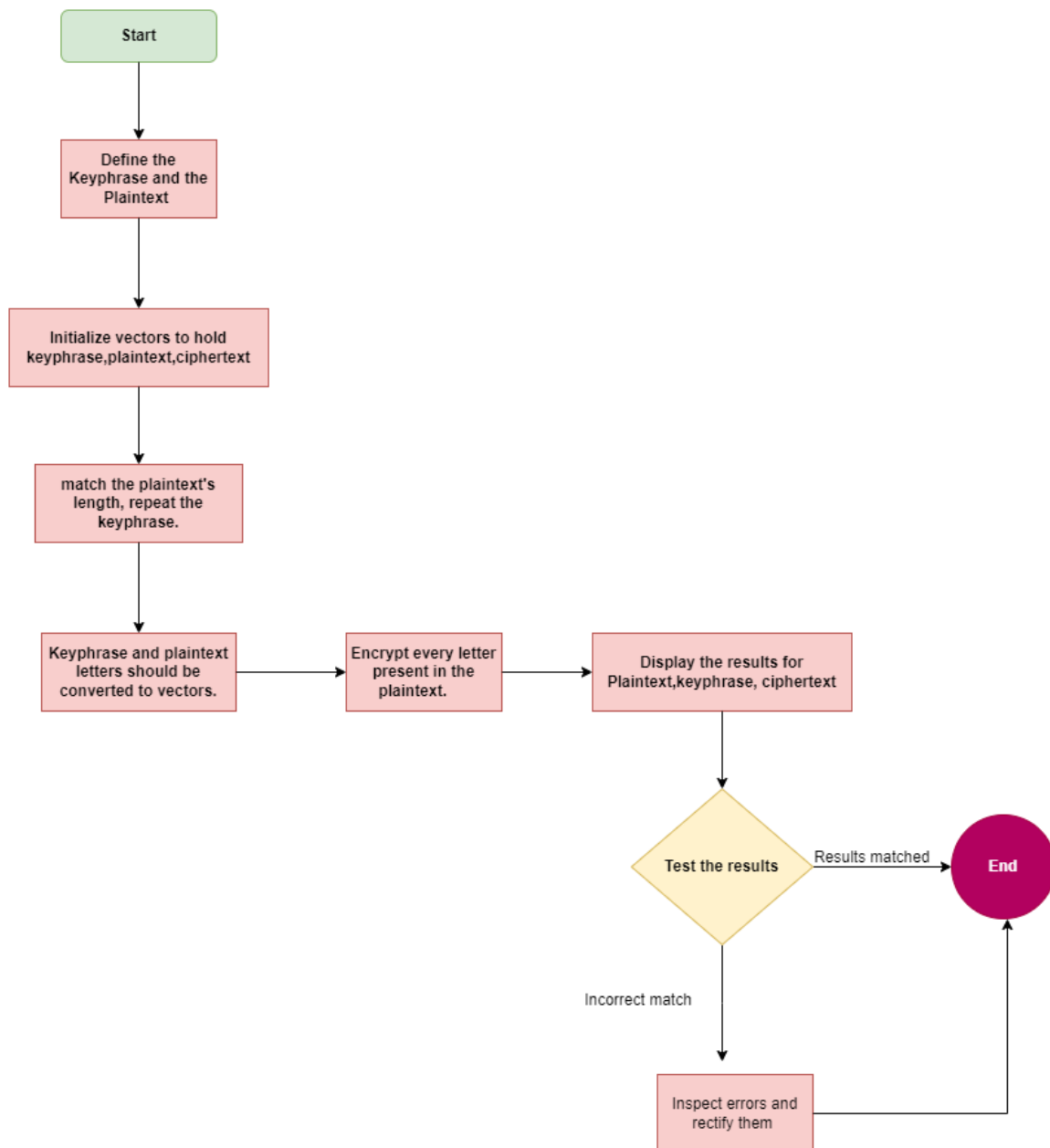
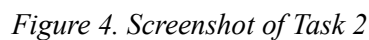


Figure 3. Flowchart for task 2



### **3. DECRYPT THE CIPHERTEXT USING THE KEYPHRASE**

#### ***Question 3.***

Decrypt the following ciphertext using the keyphrase RPROGRAMMING and store the result in the vector plaintext. ETMSXWISTBNHRIKZKKHMFGBAUDECZEEQPBBCZC.

The ciphertext is decrypted using the keyphrase in the manner described below.

**Steps 1. Initialisation:** The code starts with initialization, specifying a ciphertext ("ETMSXWISTBNHRIKZKKHMFGBAUDECZEEQPBBCZC") and a keyphrase ("RPROGRAMMING"). To hold the characters for the keyphrase, ciphertext, and decrypted plaintext, three vectors are initialised: keyphrase\_vector, ciphertext\_vector, and plaintext\_vector.

**Steps 2. Repetition of the Keyphrase:** The keyphrase is repeated until it equals the ciphertext's length. This is accomplished by appending the matching character from the keyphrase to each character in the ciphertext as iteratively as possible using a loop. To make sure the keyphrase is repeated cyclically, the modulo operation is employed.

**Steps 3. Transform Ciphertext to Numbers:** The ciphertext's letters are translated into numerical values, with 'A' standing for 0, 'B' for 1, and so on. Numerical operations are carried out in this way to facilitate decryption.

**Steps 4. Decryption process using loops:** The primary decryption loop takes the ciphertext's characters, turns them to numbers, and then deducts the keyphrase character's matching number. The outcome is then modified to make sure it is between 0 and 25 using modular arithmetic. The character value that was obtained from the encrypted numeric value is added to the plaintext vector.

**Steps 5. Show Off Results:** The original ciphertext, keyphrase, and decrypted plaintext are all shown along with the findings at the end.

To summarise, this code shows how to decipher a Vigenère cypher using a given ciphertext and keyphrase. Together with the original ciphertext and keyphrase, the generated plaintext is shown.

### Flowchart for Task 3

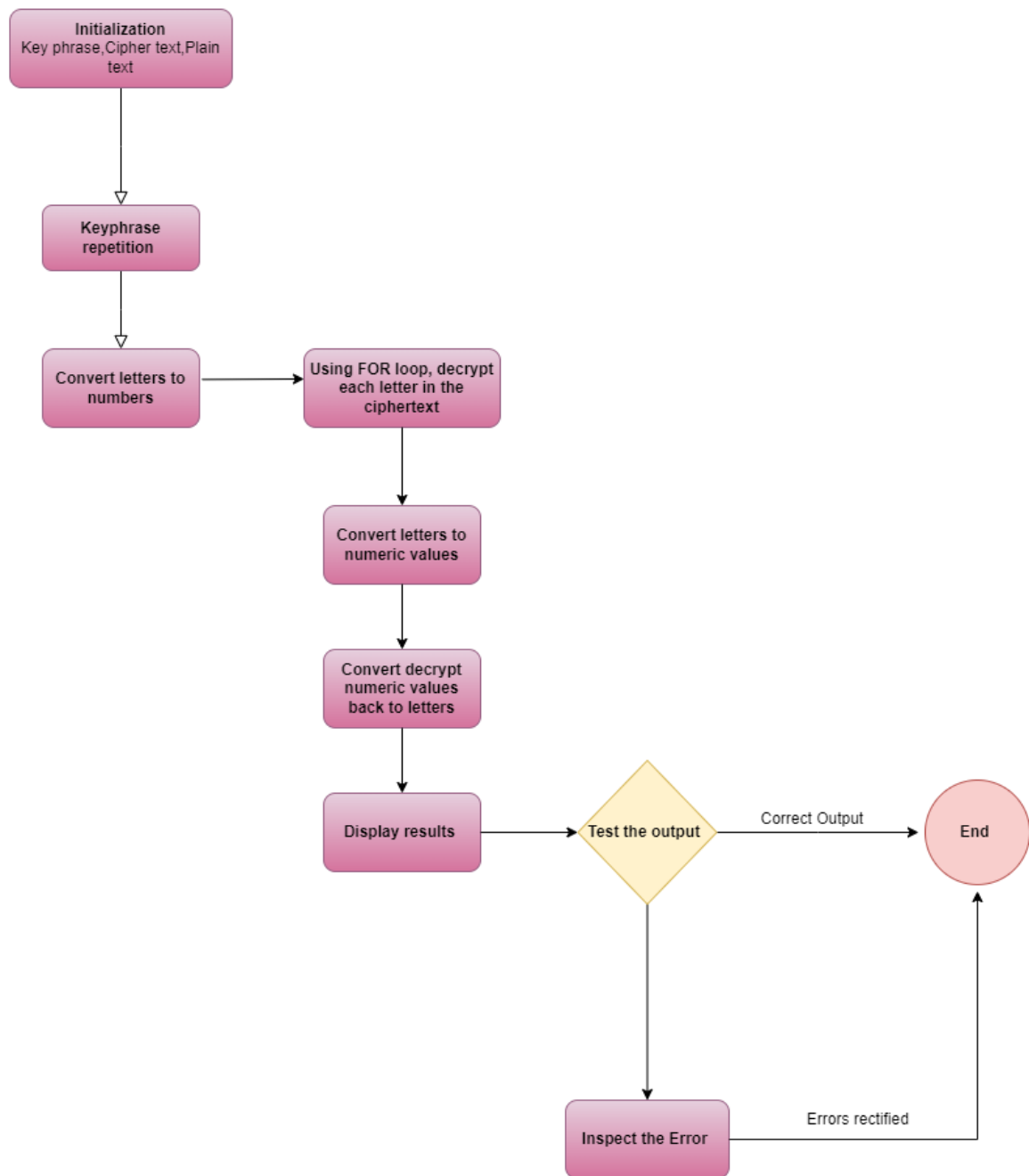


Figure 5. Flowchart for task 3

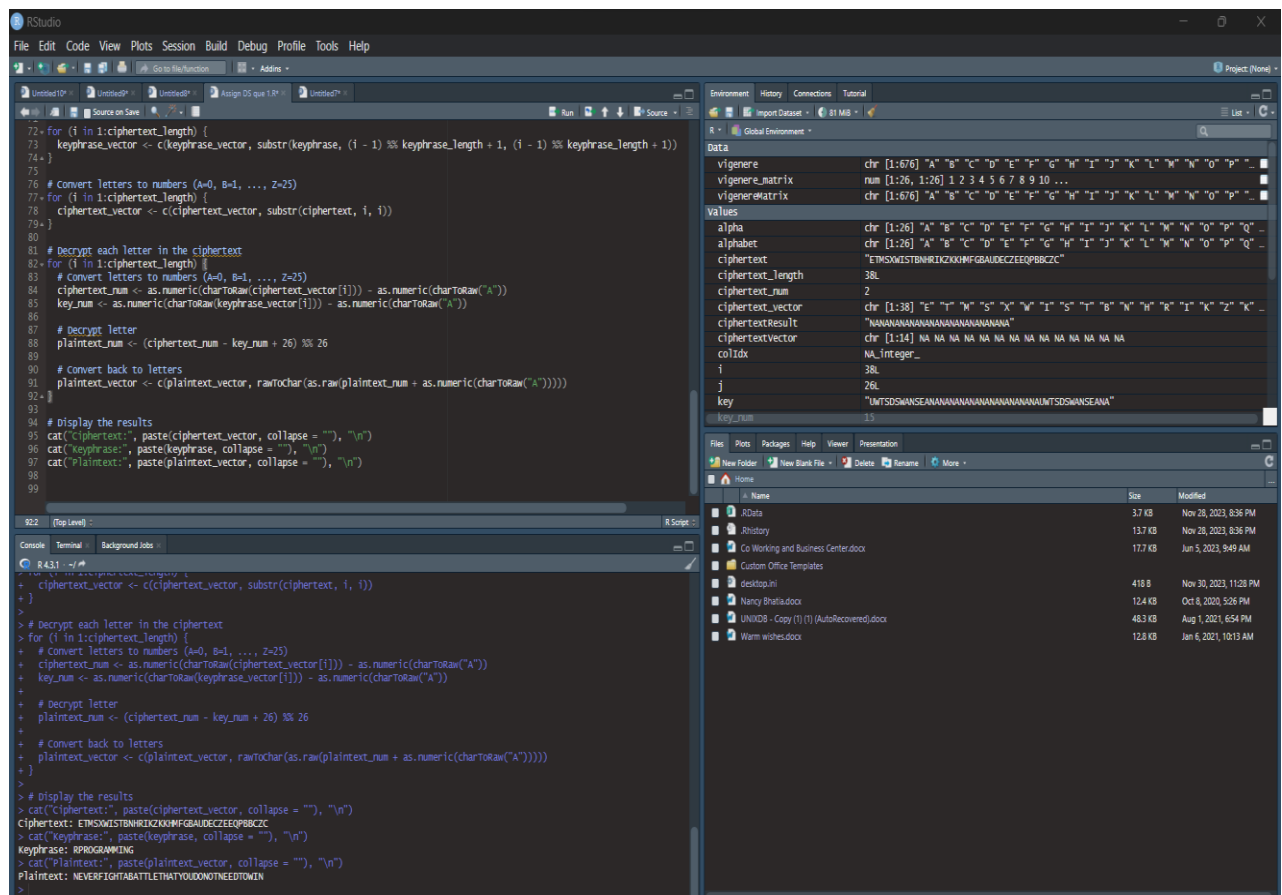


Figure 6. Screenshot of task 3

**A reflection on what you found challenging, what you learned, and what you would do differently next time.**

What I found challenging is the decryption of the ciphertext as it took few instances to work the vector. Also the code that was written in R was a bit lengthy but I managed to cope up with it. I learned that R indeed is a well written language to perform complex tasks and implement statistical programming. I would definitely try to amend the code next time into more complex yet simple queries so that it can handle more vectors and cipher heavy data. This can help to solve advanced cipher and encryption. I learned a lot about Vigenere cipher matrix as initially I wasn't well aware from it , I read a few Pdfs (which are mentioned in the references) that gave me an idea of how decryption and encryption works in Vigenere cipher, thus I was able to research more about the methods and formulas which could be helpful for encrypting or decrypting the key phrases and the ciphertext.

#### **4. QUESTIONS ASKED FROM CHAT GPT :**

1. What is The Vigenère cipher and how does it work?
2. Algorithm and logic behind Vigenère cipher matrix.
3. Formulas which can be used to obtain the Vigenère cipher matrix.
4. How does encryption takes place using a particular key phrase in Vigenère cipher matrix?
5. How to convert letter to numbers using R script code?
6. How to convert number back to letters using R script code?
7. Use of “collapse” in R script.
8. Steps for decryption in Vigenère cipher matrix.
9. Formulas that can be used in loops for decrypting a particular ciphertext using a keyphrase.
10. Use of modulo operator.

## 5. REFERENCES

1. <https://chat.openai.com/>
2. GeeksforGeeks (2016). *Vigenère Cipher - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/vigenere-cipher/>.
3. pages.mtu.edu. (n.d.). *The Vigenère Cipher Encryption and Decryption*. [online] Available at: <https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Base.html#:~:text=To%20decrypt%2C%20pick%20a%20letter> [Accessed 1 Dec. 2023].
4. The Detective Society. (n.d.). *How to Solve a Vigenere Cipher*. [online] Available at: <https://thedetectivesociety.com/how-to-solve-ciphers/vigenere-cipher/>.
5. Five Ways to Crack a Vigenère Cipher. (n.d.). Available at: <https://www.cipherchallenge.org/wp-content/uploads/2020/12/Five-ways-to-crack-a-Vigenere-cipher.pdf>.