Name: Ajay Kumar. U Date: 17/12/2020

RegNo : 312217104006 Batch - 1

(BE-CSE-'A')

CNS_SEM_PRATICALS

6. Develop a Java program to display the Vigenere matrix and implement the Vigenere Encryption

Aim :

To implement the vigenere Encryption and display the vigenere matrix.

Algorithm:

i) Read the plain-text message

ii) Read the Key

iii) Construct a Vigenere table, where each row of table consists of all letters of the English alphabet.

iv) The first row starts with the letter a and is printed till letter z for the first row, followed by next row where the letter is shifted by one letter ie., the second row starts with b and third row with c till last row which starts with z.

v) To encrypt using Vigenere cipher, We first pick a letter in the plaintext and the Corresponding letter from the key.

vi) Now, we use the key's letter as row-index and plaintext's letter as Column-index and the entry at the row-Column intersection is the letter in the Cipher text.

Vii) We follow step v) & vi) till the length of the plaintext and if the key's length is completed we again append the key with the old key for the plaintext's length.

Viii) ~~Now~~, Display the Cipher text.

Name: AjayKumar U **Class: CSE 7-A(Batch 1)**
Reg No: 312217104006 **Date: 17/12/2020**

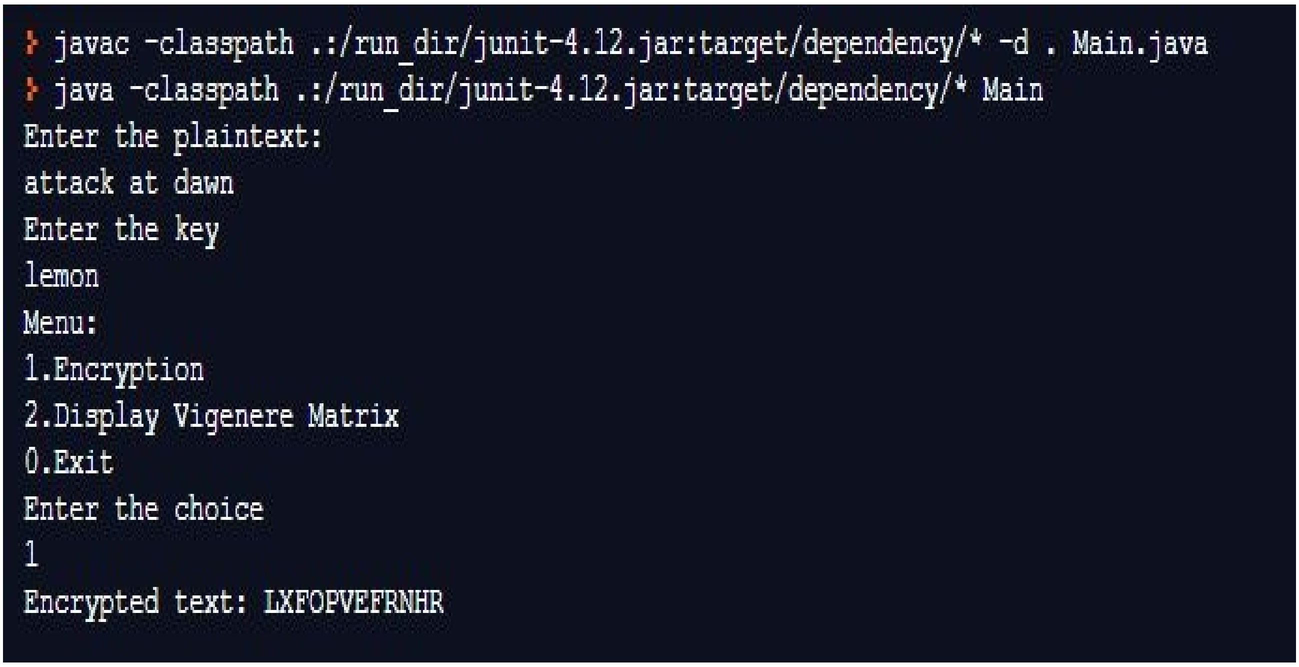
IT8761 Security Laboratory Semester Practical Exam

6. Write a program to display the Vigenere matrix and implement the Vigenere Encryption:

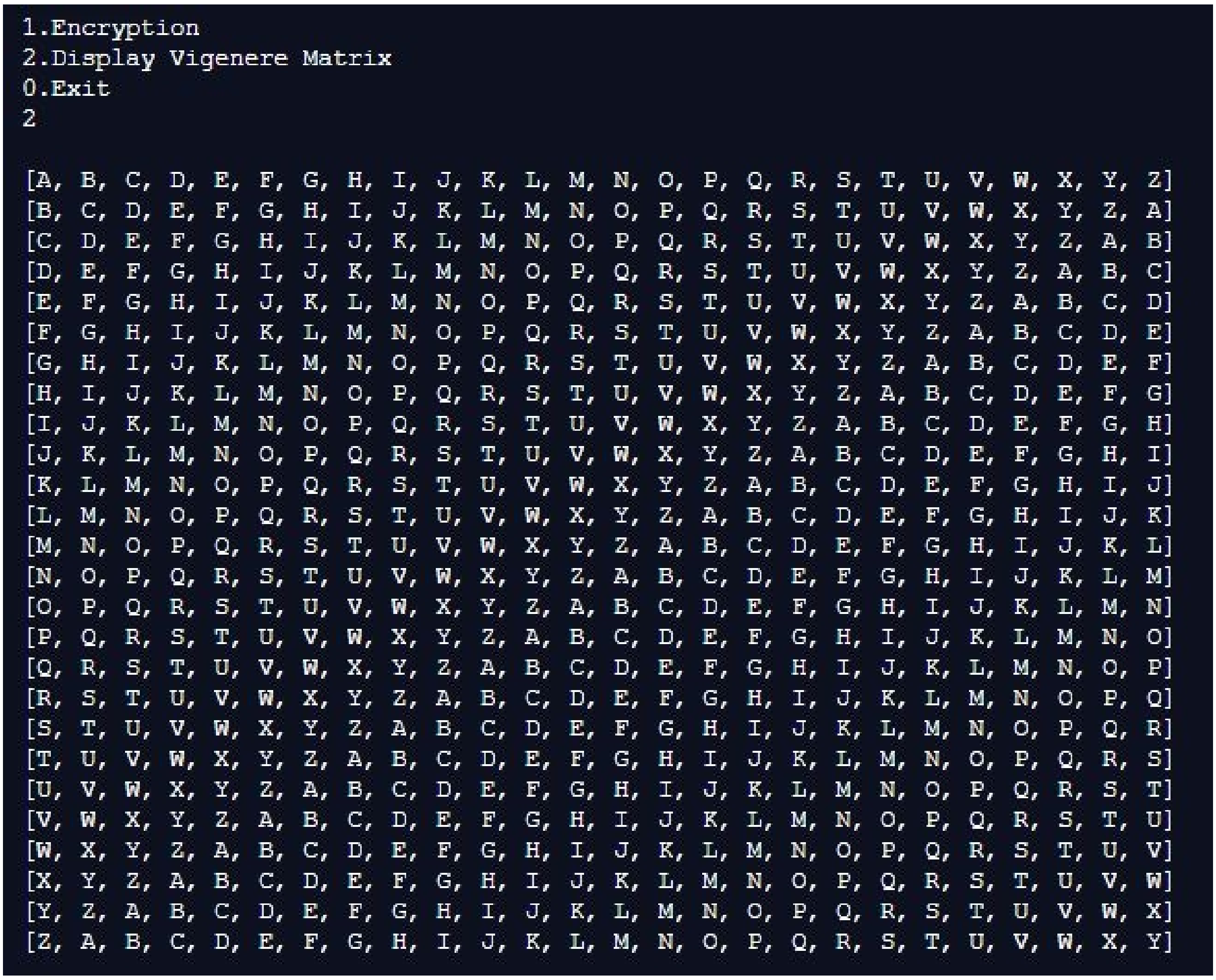
```
import java.util.*;
import java.io.*;
class Main {
  public static String Encrypt(String text,String key,char[][]matrix,String
alphabets,int n)
  {
    String result="";
    for(int i=0;i<text.length();i++)
    {

    result=result+matrix[alphabets.indexOf(key.charAt(i))][alphabets.indexOf(t
ext.charAt(i))];
    }
        return result;
  }
  public static void printmatrix(char[][]matrix,int n)
  {
    for(char[]row:matrix)
          System.out.println(Arrays.toString(row));
  }
  public static void main(String[] args) {
    Scanner scan=new Scanner(System.in);
    System.out.println("Enter the plaintext:");
    String text=scan.nextLine();
    text=text.replace(" ","");
    System.out.println("Enter the key");
    String key=scan.nextLine();
    int d=text.length()-key.length();
    for(int i=0;i<d;i++)
    {
      key=key+key.charAt(i%key.length());
    }
    String alphabets="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    int n=26,x=0;
    char [][]matrix=new char[n][n];
    for(int i=0;i<n;i++)
    {
        int k=x;
        for(int j=0;j<n;j++)
        {
            if(k==n)
              k=0;
            matrix[i][j]=alphabets.charAt(k);
            k=k+1;
        }
        x=x+1;
    }
    String cipher="";
    System.out.println("Menu:");
    System.out.println("1.Encryption");
    System.out.println("2.Display Vigenere Matrix");
    System.out.println("0.Exit");
    System.out.println("Enter the choice");
    int ch=scan.nextInt();
    while(ch!=0)
    {
        switch(ch)
        {
    case 1:
    cipher = Encrypt(text.toUpperCase(),key.toUpperCase(),matrix,alphabets,n);
    System.out.println("Encrypted text: "+cipher);
         break;
    case 2:printmatrix(matrix,n);
          break;
    default:
            System.out.println("Invalid choice");
        }
        System.out.println();
        System.out.println("Menu:");
        System.out.println("1.Encryption");
        System.out.println("2.Display Vigenere Matrix");
        System.out.println("0.Exit");
        ch=scan.nextInt();
        System.out.println();
    }
  }
}
```

Output:

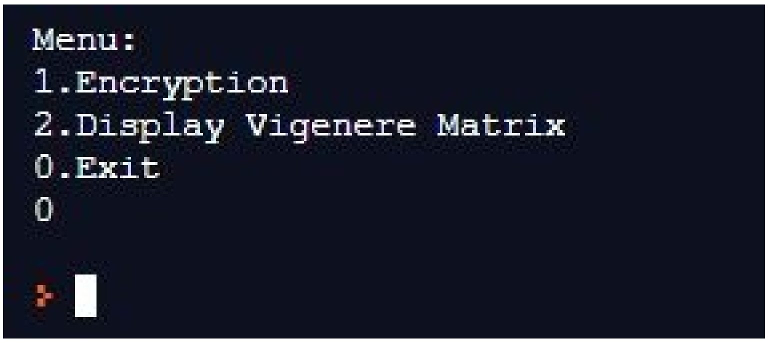


```
› javac -classpath .:/run_dir/junit-4.12.jar:target/dependency/* -d . Main.java
› java -classpath .:/run_dir/junit-4.12.jar:target/dependency/* Main
Enter the plaintext:
attack at dawn
Enter the key
lemon
Menu:
1.Encryption
2.Display Vigenere Matrix
0.Exit
Enter the choice
1
Encrypted text: LXFOPVEFRNHR
```



```
1.Encryption
2.Display Vigenere Matrix
0.Exit
2

[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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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]
```



```
Menu:
1.Encryption
2.Display Vigenere Matrix
0.Exit
0

> 
```

Result:

Vigenere Matrix is displayed and Vigenere cipher encryption is implemented.