


## MỤC LỤC

|  |    |
|--|----|
| BÀI 2: MÃ HÓA ĐỐI XỨNG CỔ ĐIỂN .....                       | 2  |
| 1.1. Thuật toán mã hóa Caesar.....                         | 2  |
| 1.2. Thuật toán mã hóa thay thế (Substitution Cipher)..... | 4  |
| 1.3. Thuật toán mã hóa Affin .....                         | 6  |
| 1.4. Thuật toán mã hóa Hill .....                          | 8  |
| 1.5. Thuật toán mã hóa hoán vị.....                        | 11 |
| 1.6. Bài tập thực hành .....                               | 11 |

|  |   |   |
|--|---|---|
| Trường ĐH CNTP TP.HCM<br>Khoa: Công nghệ thông tin<br>Bộ môn: Hệ thống thông tin<br>Môn: TH Mã hóa và ứng dụng | <b>BÀI 2:</b><br><b>MÃ HÓA ĐỐI XỨNG CỔ ĐIỂN</b> |  |
|--|---|---|

## A. MỤC TIÊU

Sau khi học xong bài này người học có khả năng:

- Cài đặt và thực thi được các thuật toán mã hóa đối xứng cổ điển bằng ngôn ngữ lập trình C#.
- Cài đặt được chương trình sử dụng mã hóa đối xứng cổ điển bằng ngôn ngữ lập trình C#.
- Đánh giá độ phức tạp và độ an toàn của các thuật toán mã hóa đối xứng cổ điển. Từ đó đưa ra chiến lược mã hóa trong thực tế.

## B. NỘI DUNG

### 1.1. Thuật toán mã hóa Caesar

#### ➤ Tóm tắt lý thuyết

- Thuật toán mã hóa thay thế mỗi chữ trong bản tin bằng chữ đứng sau nó  $k$  vị trí trong bảng chữ cái. Giả sử chọn  $k = 3$ , ta có bảng chuyển đổi như sau:

Chữ ban đầu: 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

Chữ thay thế: 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

- Mỗi chữ cái được gán một số nguyên từ 0 đến 25:

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

- Phương pháp Caesar được biểu diễn như sau: Với mỗi chữ cái  $p$  thay bằng chữ mã hóa  $C$ , trong đó:  $C = (p + k) \bmod 26$ ; và quá trình giải mã ngược lại là:  $p = (C - k) \bmod 26$ .

#### ➤ Thuật toán

Cho  $P = C = K = \mathbb{Z}_n$

Với mỗi khóa  $k \in K$ , định nghĩa:

$$e_k(x) = (x + k) \bmod n \text{ và } d_k(y) = (y - k) \bmod n \text{ với } x, y \in \mathbb{Z}_n$$

$$E = \{e_k, k \in K\} \text{ và } D = \{d_k, k \in K\}$$

- **Ví dụ 1:** Thực hiện mã hóa và giải mã theo thuật toán mã hóa Caesar với Ciphertext và key = 3 với Output như sau:

**Ciphertext:** "Wkh txlfn eurzq ira mxpsv ryhu wkh odcb grj"

**PlainText:** "The quick brown fox jumps over the lazy dog"

Để thực hiện cài đặt thuật toán mã hóa Caesar cho một chuỗi ký tự, thực hiện:

B1: Kiểm tra và chuyển ký tự về in hoa

B2: Viết hàm mã hóa

B3: Viết hàm giải mã

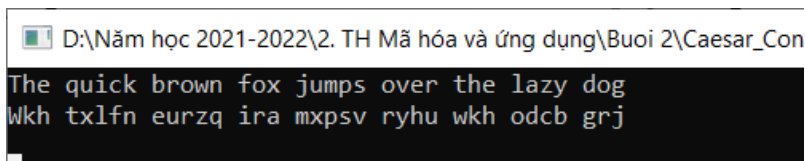
B4: Gọi hàm

```

7  namespace Caesar_Con
8  {
9      0 references
10     class Program
11     {
12         1 reference
13         private static char Cipher(char ch, int key)
14         {
15             if (!char.IsLetter(ch))
16                 return ch;
17             char offset = char.IsUpper(ch) ? 'A' : 'a';
18             return (char)((((ch + key) - offset) % 26) + offset);
19         }
20         2 references
21         public static string Encipher(string input, int key)
22         {
23             string output = string.Empty;
24             foreach (char ch in input)
25                 output += Cipher(ch, key);
26             return output;
27         }
28         1 reference
29         public static string Decipher(string input, int key)
30         {
31             return Encipher(input, 26 - key);
32         }
33         0 references
34         static void Main(string[] args)
35         {
36             string text = "The quick brown fox jumps over the lazy dog";
37             string cipherText = Encipher(text, 3);
38             string plainText = Decipher(cipherText, 3);
39             Console.WriteLine(plainText);
40             Console.WriteLine(cipherText);
41             Console.ReadLine();
42         }
43     }
44 }

```

## Kết quả



```

D:\Năm học 2021-2022\2. TH Mã hóa và ứng dụng\Buoi 2\Caesar_Con
The quick brown fox jumps over the lazy dog
Wkh txlfn eurzq ira mxpsv ryhu wkh odcg grj

```

## 1.2. Thuật toán mã hóa thay thế (Substitution Cipher)

### ➤ Tóm tắt lý thuyết

Phương pháp này thực hiện việc mã hóa thông điệp bằng cách hoán vị các phần tử trong bảng chữ cái hay tổng quát hơn là hoán vị các phần tử trong tập nguồn  $P$

### ➤ Thuật toán

Cho  $P = C = \mathbb{Z}_n$

$K$  là tập hợp tất cả các hoán vị của  $n$  phần tử  $0, 1, \dots, n-1$ . Như vậy, mỗi khóa  $\pi \in K$  là một hoán vị của  $n$  phần tử  $0, 1, \dots, n-1$ .

Với mỗi khóa  $\pi \in K$ , định nghĩa:

$$e_{\pi}(x) = \pi(x) \text{ và } d_{\pi}(y) = \pi^{-1}(y) \text{ với } x, y \in \mathbb{Z}_n$$

$$E = \{e_{\pi}, \pi \in K\} \text{ và } D = \{D_{\pi}, \pi \in K\}$$

- **Ví dụ 2:** Thực hiện mã hóa và giải mã theo thuật toán mã hóa thay thế với Ciphertext, PlainText và CipherAlphabet như sau:

**Ciphertext:** "Wkh txlfn eurzq ira mxpsv ryhu wkh odc b grj"

**PlainText:** "The quick brown fox jumps over the lazy dog"

**CipherAlphabet = "yhkqgvxfoluapwmtzecjdb snri"**

Để thực hiện cài đặt thuật toán mã hóa thay thế cho một chuỗi ký tự, thực hiện:

B1: Viết hàm mã hóa

B2: Viết hàm giải mã

B3: Gọi hàm

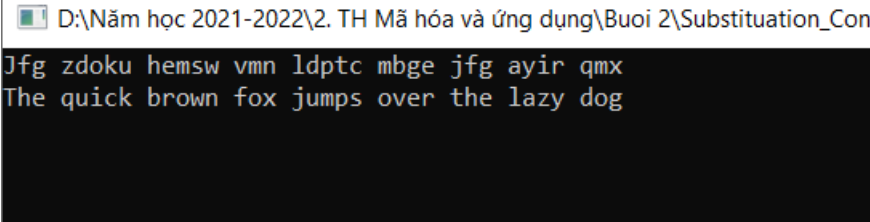
```
6
7 namespace Substitution_Cons
8 {
9     class Program
10    {
11        private static bool Cipher(string input, string oldAlphabet, string newAlphabet, out string output)
12        {
13            output = string.Empty;
14
15            if (oldAlphabet.Length != newAlphabet.Length)
16                return false;
17
18            for (int i = 0; i < input.Length; ++i)
19            {
20                int oldCharIndex = oldAlphabet.IndexOf(char.ToLower(input[i]));
21
22                if (oldCharIndex >= 0)
23                    output += char.IsUpper(input[i]) ? char.ToUpper(newAlphabet[oldCharIndex]) : newAlphabet[oldCharIndex];
24                else
25                    output += input[i];
26            }
27
28            return true;
29        }
30
31        1 reference
32        public static bool Encipher(string input, string cipherAlphabet, out string output)
33        {
34            string plainAlphabet = "abcdefghijklmnopqrstuvwxyz";
35            return Cipher(input, plainAlphabet, cipherAlphabet, out output);
36        }
37    }
38 }
```

```

36
37 1 reference
38 public static bool Decipher(string input, string cipherAlphabet, out string output)
39 {
40     string plainAlphabet = "abcdefghijklmnopqrstuvwxyz";
41     return Cipher(input, cipherAlphabet, plainAlphabet, out output);
42 }
43 0 references
44 static void Main(string[] args)
45 {
46     string text = "The quick brown fox jumps over the lazy dog";
47     string cipherAlphabet = "yhkqgvxfoluapwmtzecjdbnsri";
48     string cipherText;
49     string plainText;
50
51     bool encipherResult = Encipher(text, cipherAlphabet, out cipherText);
52     bool decipherResult = Decipher(cipherText, cipherAlphabet, out plainText);
53     Console.WriteLine(cipherText);
54     Console.ReadKey();
55 }
56 }
57 }
58 }
59

```

## Kết quả



```

D:\Năm học 2021-2022\2. TH Mã hóa và ứng dụng\Buoi 2\Substitution_Con
Jfg zdoku hemsw vmn ldptc mbge jfg ayir qmx
The quick brown fox jumps over the lazy dog

```

### 1.3. Thuật toán mã hóa Affin

- **Tóm tắt lý thuyết:** Phương pháp Affine lại là một trường hợp đặc biệt khác của mã hóa bằng thay thế mà trong đó sử dụng 2 khóa  $K_1, K_2$ .
- **Thuật toán**

Cho  $P = C = \mathbb{Z}_n$

$$K = \{(a, b) \in \mathbb{Z}_n \times \mathbb{Z}_n : \gcd(a, n) = 1\}$$

Với mỗi khóa  $k = (a, b) \in K$ , định nghĩa:

$$e_k(x) = (ax + b) \bmod n \quad \text{và} \quad d_k(x) = (a^{-1}(y - b)) \bmod n \quad \text{với } x, y \in \mathbb{Z}_n$$

$$E = \{e_k, k \in K\} \quad \text{và} \quad D = \{d_k, k \in K\}$$

- **Ví dụ 3:** Thực hiện mã hóa và giải mã theo thuật toán mã hóa Affine với PlainText, Ciphertext,  $k_1 = 3, k_2 = 2$  được mô tả như sau:  
**Ciphertext:** "Wkh txlfn eurzq ira mxpsv ryhu wkh odc b grj"

**PlainText:** "The quick brown fox jumps over the lazy dog"

**Các bước thực hiện:**

B1: Viết hàm mã hóa: AffineEncrypt

B2: Viết hàm giải mã: AffineDecrypt

B3: Viết hàm lấy nghịch đảo của số nguyên mod 26:

MultiplicativeInverse.

---

```
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace Affine_Cons
8  {
9      0 references
10     class Program
11     {
12         1 reference
13         public static string AffineEncrypt(string plainText, int a, int b)
14         {
15             string cipherText = "";
16
17             // Put Plain Text (all capitals) into Character Array
18             char[] chars = plainText.ToUpper().ToCharArray();
19
20             // Compute e(x) = (ax + b)(mod m) for every character in the Plain Text
21             foreach (char c in chars)
22             {
23                 int x = Convert.ToInt32(c - 65);
24                 cipherText += Convert.ToChar(((a * x + b) % 26) + 65);
25             }
26
27             return cipherText;
28         }
29
30         /// This function takes cipher text and decrypts it using the Affine Cipher
31         /// d(x) = aInverse * (e(x) - b)(mod m).
32         1 reference
33         public static string AffineDecrypt(string cipherText, int a, int b)
34         {
35             string plainText = "";
36
37             // Get Multiplicative Inverse of a
38             int aInverse = MultiplicativeInverse(a);
```

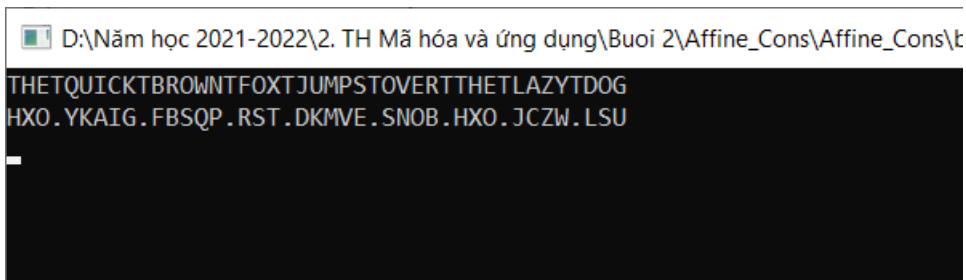
---

```

35
36 // Put Cipher Text (all capitals) into Character Array
37 char[] chars = cipherText.ToUpper().ToCharArray();
38
39 // Computer d(x) = aInverse * (e(x) - b)(mod m)
40 foreach (char c in chars)
41 {
42     int x = Convert.ToInt32(c - 65);
43     if (x - b < 0) x = Convert.ToInt32(x) + 26;
44     plainText += Convert.ToChar(((aInverse * (x - b)) % 26) + 65);
45 }
46
47 return plainText;
48 }
49 /// This functions returns the multiplicative inverse of integer a mod 26.
    1 reference
50 public static int MultiplicativeInverse(int a)
51 {
52     for (int x = 1; x < 27; x++)
53     {
54         if ((a * x) % 26 == 1)
55             return x;
56     }
57     throw new Exception("No multiplicative inverse found!");
58 }
    0 references
59 static void Main(string[] args)
60 {
61     string text = "The quick brown fox jumps over the lazy dog";
62     string cipherText = AffineEncrypt(text, 3, 2);
63     string plainText = AffineDecrypt(cipherText, 3, 2);
64     Console.WriteLine(plainText);
65     Console.WriteLine(cipherText);
66     Console.ReadLine();
67 }
68 }
69 }

```

**Kết quả:**



```

D:\Năm học 2021-2022\2. TH Mã hóa và ứng dụng\Buoi 2\Affine_Cons\Affine_Cons\
THETQUICKTBROWNTFOXJTJUMPSTOVERTTTHETLAZYTDG
HXO.YKAIG.FBSQP.RST.DKMVE.SNOB.HXO.JCZW.LSU

```

#### 1.4. Thuật toán mã hóa Hill

- **Tóm tắt lý thuyết:** Phương pháp Hill được Lester S. Hill công bố năm 1929: Cho số nguyên dương  $m$ , định nghĩa  $P = C = (\mathbb{Z}n)^m$ . Mỗi phần tử  $x \in P$  là một bộ  $m$  thành phần, mỗi thành phần thuộc  $\mathbb{Z}n$ . Ý tưởng chính của phương pháp này là sử dụng  $m$  tổ hợp tuyến tính của  $m$  thành phần trong mỗi phần tử  $x \in P$  để phát sinh ra  $m$  thành phần tạo thành phần tử  $y \in C$ .

- **Thuật toán**



Chọn số nguyên dương  $m$ . Định nghĩa:

$P = C = (\mathbb{Z}_n)^m$  và  $K$  là tập hợp các ma trận  $m \times m$  khả nghịch

Với mỗi khóa  $k = \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & \cdots & \cdots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix} \in K$ , định nghĩa:

$$e_k(x) = xk = (x_1, x_2, \dots, x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & \cdots & \cdots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix} \text{ với } x = (x_1, x_2, \dots, x_m) \in P$$

và  $d_k(y) = yk^{-1}$  với  $y \in C$ .

Mọi phép toán số học đều được thực hiện trên  $\mathbb{Z}_n$ .

- **Ví dụ 4:** Minh họa thuật toán mã hóa Hill: Nhập chuỗi ký tự, thực hiện mã hóa, và xuất kết quả. (Sinh viên thực hiện quy trình giải mã)

```

1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace Hill_Cons
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int i, j, sum = 0, end = 0;
16             int[,] mtrx = new int[25, 25];
17             int[,] ans = new int[25, 1];
18             string text = "";
19
20             Console.WriteLine("Enter your Plaintext");
21             Console.Write("\n");
22             text = Console.ReadLine();
23             Console.Write("\n");
24             char[] txt = text.ToCharArray();
25             end = txt.Length;
26             for (i = 0; i < end; i++)
27             {
28                 txt[i] = Convert.ToChar(txt[i] - 'a');
29             }
30             Random rnd = new Random();
31             for (i = 0; i < end; i++)
32             {
33                 for (j = 0; j < end; j++)
34                 {
35                     mtrx[i, j] = rnd.Next();
36                 }
37             }
38             for (i = 0; i < end; i++)
39             {
40                 sum = 0;
41                 for (j = 0; j < end; j++)
42                 {
43                     sum += mtrx[i, j] * (int)txt[j];
44                 }
45                 ans[i, 0] = sum;
46             }
47
48             Console.Write("Your CipherText is:");
49             for (i = 0; i < end; i++)
50             {
51                 char cipher = (char)((((ans[i, 0]) % 26) + 97));
52                 Console.Write("\t" + cipher);
53             }
54
55             Console.ReadKey();
56
57         }
58     }
59 }
60
61
62
63

```

## Kết quả:

```
D:\Năm học 2021-2022\2. TH Mã hóa và ứng dụng\Buoi 2\Hill_Cons\Hill_Cons\bin\
Enter your Plaintext
hill
Your CipherText is:  l      r      P      u
```

### 1.5. Thuật toán mã hóa hoán vị

- **Tóm tắt lý thuyết:** Thay thế mỗi ký tự trong thông điệp nguồn bằng một ký tự khác để tạo thành thông điệp đã được mã hóa. Ý tưởng chính của phương pháp mã hóa hoán vị (Permutation Cipher) là vẫn giữ nguyên các ký tự trong thông điệp nguồn mà chỉ thay đổi vị trí các ký tự; nói cách khác thông điệp nguồn được mã hóa bằng cách sắp xếp lại các ký tự trong đó.

- **Thuật toán**

Chọn số nguyên dương  $m$ . Định nghĩa:

$P = C = (\mathbb{Z}_n)^m$  và  $K$  là tập hợp các hoán vị của  $m$  phần tử  $\{1, 2, \dots, m\}$

Với mỗi khóa  $\pi \in K$ , định nghĩa:

$$e_{\pi}(x_1, x_2, \dots, x_m) = (x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(m)}) \text{ và}$$

$$d_{\pi}(y_1, y_2, \dots, y_m) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \dots, y_{\pi^{-1}(m)})$$

với  $\pi^{-1}$  hoán vị ngược của  $\pi$

- **Ví dụ 5:** Cài đặt thuật toán mã hóa hoán vị, nhập vào chuỗi Text và xuất kết quả sau khi thực hiện mã hóa chuỗi này.

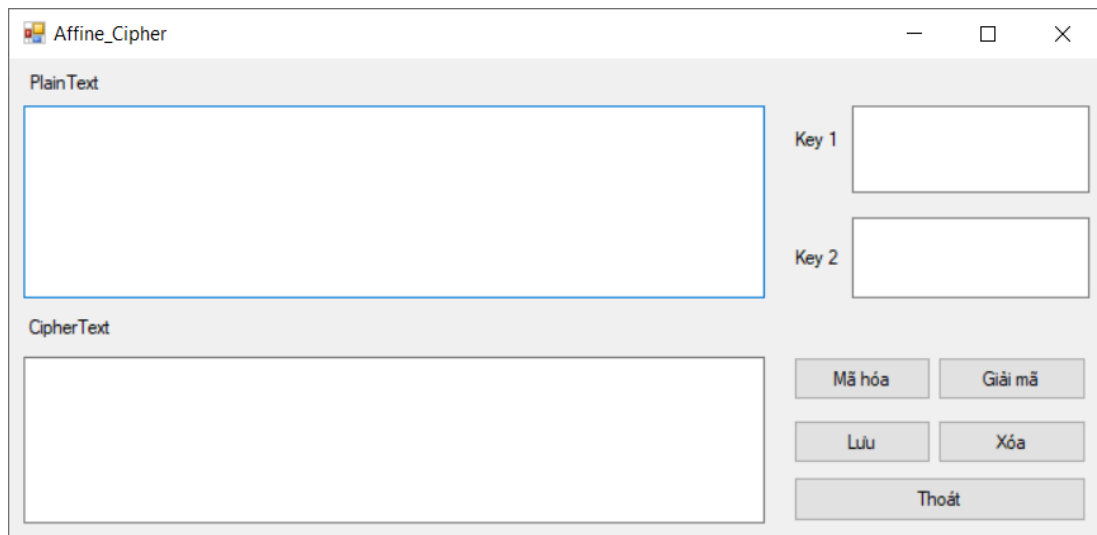
### 1.6. Bài tập thực hành

- Viết chương trình giải mã bản mã sau, giả sử mã hóa Ceasar được sử dụng để mã hóa với  $k=3$ :  
IRXUVFRUHDQGVHYHQBHDUVDJR
- Viết chương trình mã hóa và giải mã một file văn bản ASCII trên máy tính bằng phương pháp mã hóa Ceasar.

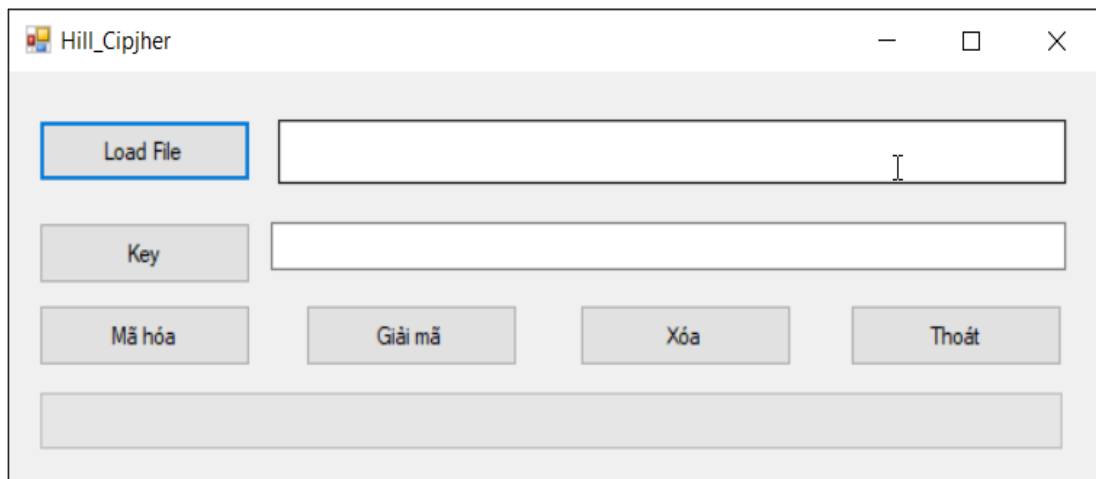
3. Viết chương trình thực hiện mã hóa Caesar với giao diện như sau:

4. Viết chương trình thực hiện mã hóa bản rõ sau: “enemy coming”, dùng phương pháp mã hóa thay thế với khóa  $K = \text{IAUTMOCSNREBDLHVWYFPsZJXKGQ}$
5. Viết chương trình thực hiện mã hóa thay thế với giao diện như sau. Trong đó Key người dùng tự nhập.

6. Viết chương trình thực hiện mã hóa Affine với giao diện như sau: Trong đó Key 1, key 2 người dùng tự nhập.



7. Viết chương trình thực hiện mã hóa Hill với giao diện như sau. Trong đó Key người dùng tự nhập.



8. Viết chương trình thực hiện mã hóa thông điệp sau bằng phương pháp hoán vị:  
 PlainText = "we are all together"  
 biết khóa  $k = 24153$
9. Viết chương trình thực hiện mã hóa Hoán vị với tập tin được load bất kỳ.