

維吉尼亞密碼

計算機程式概論 作業三 書面報告

余京儒 心理四 b04310036

目錄

- 一、 動機
- 二、 構想解說與程式測試及規劃
- 三、 流程圖
- 四、 程式測試執行結果
- 五、 參考資料
- 六、 程式列表

一、動機

這次的作業是延續之前的作業二。目標是想要把上次沒有寫成功成類別的東西，在改寫成類別，與將程式碼改得更加簡潔。並融合凱薩密碼與維吉尼亞密碼，也就是使用這可以使用單個數字、一串數字或者字母來做為金鑰。結果都可以將之加密或解密。

二、構想解說與程式測試及規劃

維吉尼亞密碼是由一些偏移量不同的凱薩密碼所組成，一般的加密方法如下：

假設明文為 ATTACKATDAWN，金鑰為 lemon，先重複金鑰至與明文一樣長度，再根據維吉尼亞方格，將明文的第 k 個字，對應以金鑰的第 k 個字作為開頭的維吉尼亞方格中的行做轉換，如下：

明文: ATTACKATDAWN

金鑰: LEMONLEMONLE

密碼: LXFOPVEFRNHR

	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

圖片來源: 維基百科

在我的這次程式中，金鑰可以是任意長度的英文或數字。倘若是數字的話，也會跟英文一樣將它重複至與明文一樣長並一對一進行位移。我的類別有下列四種：

1. **IfError**: iferror() 用來確認輸入的明文是否都只包含英文字母，與 iferrorKEY() 用來確認金鑰是否都只包含英文字母或數字，若含有其他字母則重新輸入。
2. **Repkey**: repkey() 用來將金鑰的重複到跟明文一樣長度。
3. **ASCII**: 用來將明文與金鑰轉換成 ASCII。因為金鑰其實為對應順序明文的位移量，因此分成兩個功能來寫。KeyToASCII() 是將金鑰轉為位移量後在做大小寫的處理，讓不論大寫還小寫都是一樣的位移量。而 ToASCII 則是負責將明文轉成對應的 ASCII。
4. **WordtoCode**: 含有三個成員函式。其中 run() 是參考老師的範例 calculator 的寫法，將運作寫在成員函式中。。Encrypt() 為加密，將明文加上金鑰後取除以 26 的餘數，再轉回字元。Decrypt() 為解密，將明文扣除金鑰後加 26，取除以 26 的餘數，再轉回字元。

三、程式流程圖(20%)

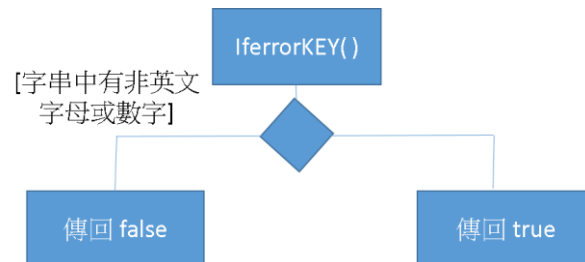
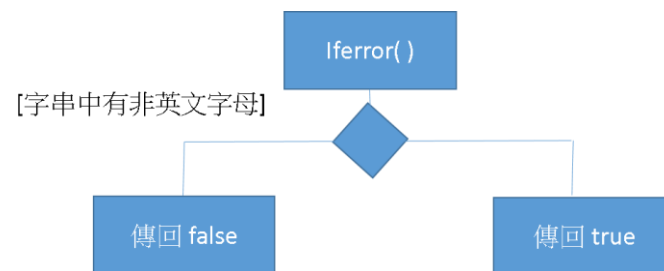
i. Main



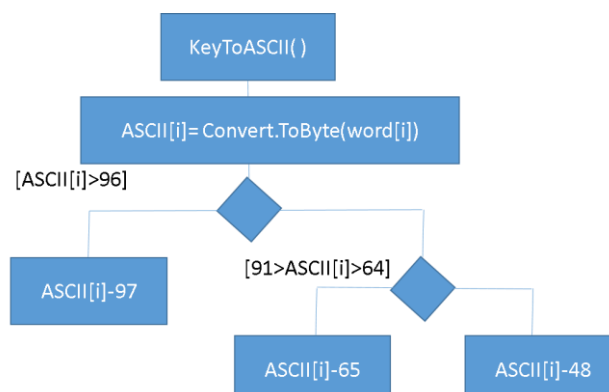
ii. RepKey

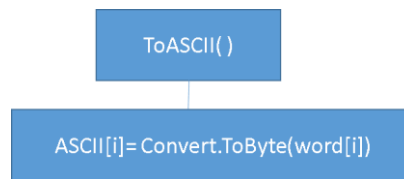


iii. ifError

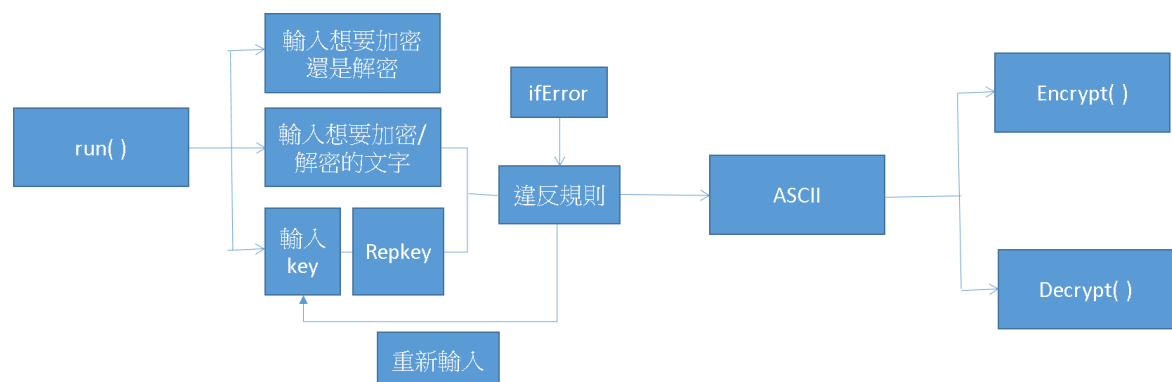
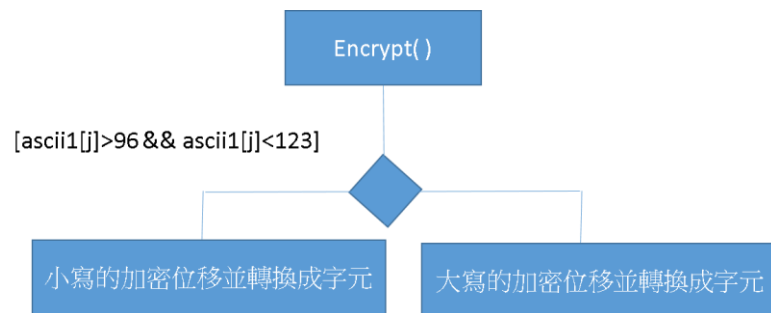
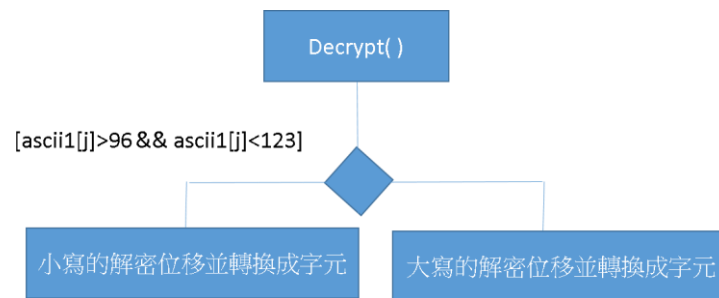


IV. ASCII



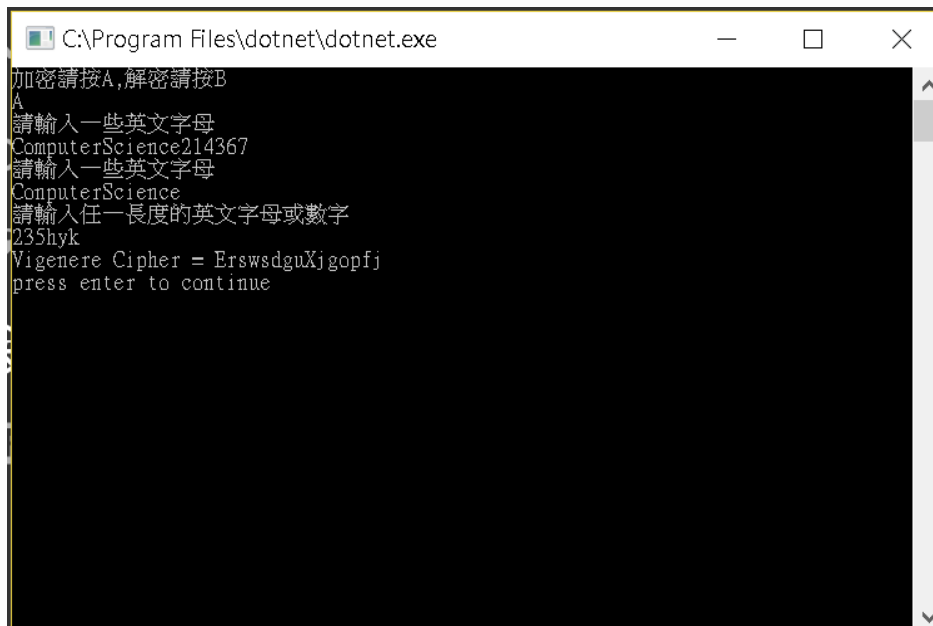


V. WordtoCode



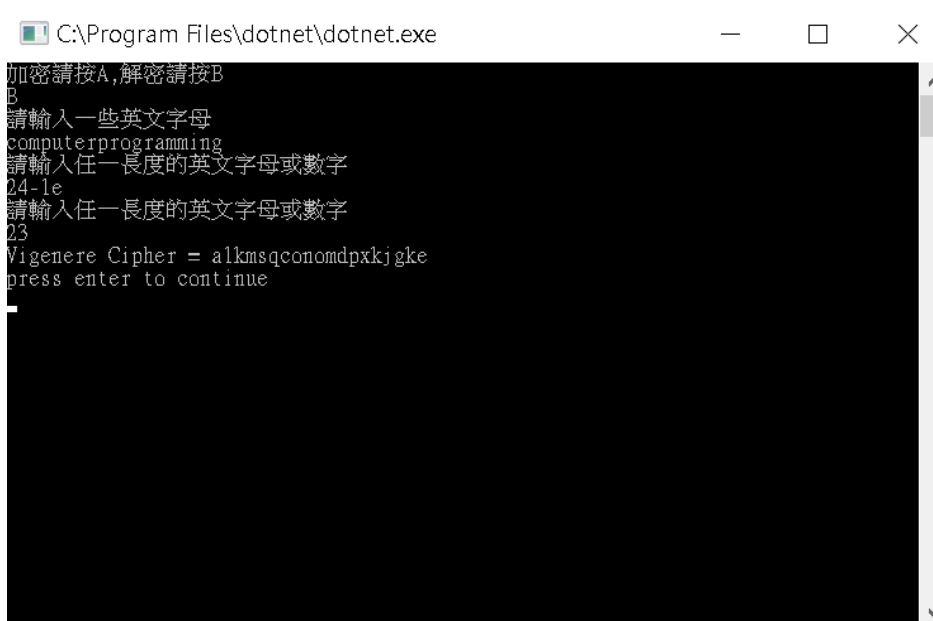
四、程式測驗執行結果

1. 加密



```
C:\Program Files\dotnet\dotnet.exe
加密請按A,解密請按B
A
請輸入一些英文字母
ComputerScience214367
請輸入一些英文字母
ComputerScience
請輸入任一長度的英文字母或數字
235hyk
Vigenere Cipher = ErswsdguXjgopfj
press enter to continue
```

2. 解密



```
C:\Program Files\dotnet\dotnet.exe
加密請按A,解密請按B
B
請輸入一些英文字母
computerprogramming
請輸入任一長度的英文字母或數字
24-le
請輸入任一長度的英文字母或數字
23
Vigenere Cipher = alkmsqconomdpkjkge
press enter to continue
```

五、參考資料

1. C#上課講義
2. 維基百科—維吉尼亞密碼:
<https://zh.wikipedia.org/wiki/%E7%BB%B4%E5%90%89%E5%B0%BC%E4%BA%9A%E5%AF%86%E7%A0%81>

3. 維基百科—ASCII: <https://zh.wikipedia.org/wiki/ASCII>

六、程式列表

program.cs

```
using System;

namespace HW3
{
    class Program
    {
        static void Main(string[] args)
        {
            WordtoCode Q = new WordtoCode();
            Q.run();

            Console.WriteLine('\n'+ "press enter to continue");
            Console.ReadLine();
        }
    }
}
```

RepKey

```
using System;

namespace HW3
{
    public class RepKey
    {
        public static string repkey(string word, string key)
        {
            System.Text.StringBuilder key1 = new System.Text.StringBuilder(key);
            for (int i=0; i<word.Length; i++)
            {
                key1.Append(key);
            }
            return(key1.ToString());
        }
    }
}
```

```
    }  
}
```

IfError.cs

```
using System;  
  
namespace HW3  
{  
    public class IfError  
    {  
  
        public static bool iferror(byte[] a)  
        {  
            for (int j =0; j< a.Length; j++)  
            {  
                if ((a[j]>64 && a[j]<91) || (a[j]>96 && a[j]<123))  
                {  
                    continue; //處理非英文字母  
                }  
                else  
                {  
                    return false;  
                }  
            }  
            return true;  
        }  
  
        public static bool iferrorKEY(byte[] a)  
        {  
            for (int j =0; j< a.Length; j++)  
            {  
                if ((a[j]>64 && a[j]<91) || (a[j]>96 && a[j]<123) || (a[j]>47 && a[j]<58))  
                {  
                    continue; //處理非英文字母  
                }  
                else  
                {  
                    return false;  
                }  
            }  
            return true;  
        }  
    }  
}
```



```

        }
    }
    return true;
}

}
}

```

ASCII.cs

```

using System;

namespace HW3
{
    public class ASCII
    {
        public static byte[] ToASCII( string word) //將明文轉為 ASCII
        {
            byte[] ascii = new byte[word.Length];
            for (int i=0; i< word.Length; i++){
                //Console.WriteLine (chars[i]);
                char[] chars = word.ToCharArray();
                ascii[i] = Convert.ToByte(chars[i]);
            }
            return ascii;
        }

        public static byte[] KeyToASCII(string word)//將 KEY 轉為 ASCII
        {
            byte[] ascii1 = new byte[word.Length];
            for (int i=0; i< word.Length; i++){
                //Console.WriteLine (chars[i]);
                ascii1[i] = Convert.ToByte(word[i]);
                if (ascii1[i]>96){ //因為 KEY 跟大小寫沒有關係，所以做此處理
                    ascii1[i]-= 97; //將不論大寫還是小寫都改為 a=1 開始
                }
                else if (ascii1[i]>64 && ascii1[i]<91)
                {
                    ascii1[i]-=65;
                }
            }
        }
    }
}

```

```

        }
        else
        {
            ascii1[i]-=48;
        }
    }
    return ascii1;
}
}
}

```

WordtoCode.cs

```

using System;

namespace HW3
{
    public class WordtoCode
    {
        private bool e;
        private bool r ;
        protected string b ;
        protected string a ;
        private char op;

        private byte[] ascii1;
        private byte[] ascii2;

        private char[] code ;

        public void Encrypt ()
        {
            code = new char[ascii1.Length];
            for (int j =0; j< ascii1.Length; j++){
                if (ascii1[j]>96 && ascii1[j]<123){
                    code[j] = Convert.ToChar(((ascii1[j]-96+ascii2[j])%26)+96);
                } //處理小寫英文字母
                else if (ascii1[j]>64 && ascii1[j]<91){
                    code[j] = Convert.ToChar(((ascii1[j]-64+ascii2[j])%26)+64);
                } //處理大寫英文字母
            }
        }
    }
}

```

```

    }

}

public void Decrypt ()
{
    code = new char[ascii1.Length];
    for (int j =0; j< ascii1.Length; j++){
        if (ascii1[j]>96 && ascii1[j]<123)
        {
            code[j] = Convert.ToChar(((ascii1[j]-96-ascii2[j]+26)%26)+96);
        } //處理小寫英文字母
        else if (ascii1[j]>64 && ascii1[j]<91)
        {
            code[j] = Convert.ToChar(((ascii1[j]-64-ascii2[j]+26)%26)+64);
        } //處理大寫英文字母
    }
}

public void run()
{
    Console.WriteLine("加密請按 A,解密請按 B"); //輸入加密或解密
    op = Convert.ToChar(Console.ReadLine()) ;

    while(e==false) //檢查輸入的字串中是否含有非英文文字的符號，若有則重新輸入
    {
        Console.WriteLine("請輸入一些英文字母");
        a = Console.ReadLine() ;
        ascii1 = ASCII.ToASCII(a);
        e = IfError.iferror(ascii1);
    }

    while(r==false) //檢查 Key 中是否含有非英文文字的符號，若有則重新輸入
    {
        Console.WriteLine("請輸入任一長度的英文字母或數字");
        b = Console.ReadLine();
        ascii2 = ASCII.KeyToASCII(RepKey.repkey(a,b)); //呼叫涵式 repkey 將 key 重複
                                                    至與明文一樣長
        r = IfError.iferrorKEY(ASCII.ToASCII(b));
    }
}

```

```
switch(op)
{
    case 'A':
        Encrypt ();
        break;
    case 'B':
        Decrypt ();
        break;
    default:
        Console.WriteLine("error!!!");
        break;
}

Console.Write("Vigenère Cipher = ");
for(int i = 0; i< a.Length; i++)
{
    Console.Write(code[i]);
}
}
}
```