Playfair Cipher

姓名：李名智
學號：1102924

一、介紹

由組合語言來實作 Playfair Cipher，而 Playfair Cipher 是一種對稱式密碼，一種雙字母取代的加密方法。

作法：

1. 設計一個 5x5 的矩陣，矩陣內填入 A-Z（I/J 視為同一字）
2. 將輸入文字做兩兩一組。若組內相同字母，將"X"插入兩字字母之間，並重新分組（例如 HELLO 將分成 HE LX LO）。若最後剩一個字母，也加入"X"。
3. 每組中，依 Playfair Key Matrix 做相對應的操作（分為三種情況）：
   情況一：若兩字不同行、不同列，則在矩陣中找出另外兩個字母，使四個字母呈長方形。
   情況二：若兩字同橫行，則取字母的右邊一位，若最後邊字母則取最左邊的字母。
   情況三：若兩字同直列，則取字母的下方一位，若最下方字母則取最上方的字母。
4. 找到新兩字母為原字母的加密結果。

設計：Playfair Key Matrix

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

二、程式說明

(1) data

```
.data
msg1  BYTE  "Please input the plaintext: ",0
msg2  BYTE  "Modified plaintext: ",0
msg3  BYTE  "The ciphertext is: ",0
mystring   BYTE   BUFMAX+1 DUP(?) ;儲存輸入文字
myupper   BYTE   BUFMAX+1 DUP(?)  ;儲存變大寫的文字(單純文字)
myTwo   BYTE   BUFMAX+1 DUP(?)
strSize  DWORD  ?          ;輸入文字的長度

div2 DWORD 00000002h
div5 DWORD 00000005h
Playfairkey BYTE "MONARCHYBDEFGIKLPQSTUVWXZ",0
myPf   BYTE   BUFMAX+1 DUP(0)
tmp DWORD ?          ;用來存ecx
w1pos DWORD ?        ;找到字的位置
row1 DWORD ?         ;存該字在第幾行第幾列
col1 DWORD ?
w2pos DWORD ?
row2 DWORD ?
col2 DWORD ?
GO DWORD 0           ;用來判斷是否跳到該跳的位置 1.往下比 2.跳到輸出位置
```

(2) 主程式介紹

```
main PROC
    ;將參數PUSH到steak
    mov eax, OFFSET Playfairkey   ;key[ebp+16]
    push eax
    mov eax, OFFSET myTwo  ;inputstring[ebp+12]
    push eax
    mov eax, OFFSET myPf  ;output[ebp+8]
    push eax

    call Playfair

    call Crlf
    mov edx,OFFSET msg3
    call WriteString
    mov edx,OFFSET myPf
    call WriteString
    exit
main ENDP
```

(3) 副程式介紹

stack 概念

```
;-------------------------------------------------------
;Playfairkey       high    +16
;myTwo
;myPf
;return address
;舊ebp值            low     +0
;-------------------------------------------------------
```

副程式一：Playfair

（列印 msg1、msg2 內容，並 push 舊 ebp、取得參數位址）

```
Playfair PROC
    mov edx,OFFSET msg1 ;輸出enter your string
    call WriteString
    call InputTheString ;輸入文字
    call lowertoCap     ;小寫變大寫的副程式(並把所有J改成I)
    call Crlf

    call Twotwo
    mov edx,OFFSET msg2 ;
    call WriteString
    mov edx,OFFSET myTwo
    call WriteString
    call Crlf
    ;---------
    push ebp
    mov ebp,esp
    pushad
    ;---------
    ;mov edi,[ebp+16]  ;key
    mov esi,[ebp+12]  ;mytwo
    mov ebx,[ebp+8]
    mov ecx,50
L1:
```

（用迴圈判斷）

```
L1:
    mov tmp,ecx
    mov al,[esi]
    cmp al,' '    ;若為空格就輸入空格進myPf
    je spa
    cmp al,0      ;是否遇到結束符號
    je quit
    ;-------------------------第一個字
    mov edi,[ebp+16] ;key
    call Find1
   ;-------------------------第二個字
    mov al,[esi+1]
    mov edi,[ebp+16] ;key
    call Find2
```

```
   ;---------------------------比較
   ;同row 往右 (若col+1=5 wpos就-4，否則wpos+1)
CompareRow:
    call CmpRow
    mov eax,GO
    cmp GO,1
    je CompareCol
    jmp Input
   ;同col 往下 (若row+1=5 wpos就-20,否則wpos+5)
CompareCol:
    call CmpCol
    mov eax,GO
    cmp GO,1
    je Different
    jmp Input
Different:
    call Other
    jmp Input
Input:
    call InputPF
    jmp Next
 spa:
    mov al,' '
    mov [ebx], al
    inc ebx
    inc esi
 Next:
    loop L1
quit:
    ;---------
    popad
    pop ebp
    ret 12     ;add esp, 12
Playfair ENDP
```

副程式二：InputTheString(用來輸入文字)

```
InputTheString PROC
    pushad
    mov ecx,BUFMAX           ; 可以輸多長的文字
    mov edx,OFFSET mystring  ; 將輸入文字存到mystring
    call ReadString          ; 輸入
    mov strSize,eax          ; 儲存文字長度
    popad
    ret
InputTheString ENDP
```

副程式三：lowertoCap(將輸入的文字轉成大寫並將所有 J/j 都轉成 I)

```
lowertoCap PROC
    pushad
    mov esi, OFFSET mystring  ; 輸入的文字
    mov edi, OFFSET myupper
    mov ecx, strSize          ; 長度

L1: mov al, [esi]
    cmp al, 0                 ; 檢查是否到達結尾
    je do                     ; 如果是，結束循環
    cmp al,'j'                ; 將J或j改成'i'
    je upJ
    cmp al,'J'
    je upJ
    jmp Tran1
upJ:
    mov bl,'i'
    mov [esi],bl
    mov al,[esi]
Tran1:
    cmp al, 'a'               ; 檢查是否為小寫字母
    jb A1                     ; 如果小於 'a'，跳過轉換
    cmp al, 'z'
    ja A1
    sub al, 32                ; 若a<=[esi]<=z 將小寫字母轉換為大寫字母
```

```
A1: cmp al,'A'
    jb quit
    cmp al,'Z'
    ja quit
    mov [edi], al
    inc edi
quit:
    inc esi
    loop L1

do: mov al,'5'
    mov [edi], al
    mov BYTE PTR [edi+1], 0      ;確保輸出以 NULL 結尾
    popad
    ret
lowertoCap ENDP
```

副程式四：Twotwo(將處理好的文字用成兩兩一組的 array)

```
Twotwo PROC
    pushad
    mov esi,OFFSET myupper    ;處理成大寫的字串
    mov edi,OFFSET myTwo
    mov eax,strSize
    mov edx,00000000h
    div div2
    inc eax

    mov ecx,eax
L1: mov al,[esi]      ;若讀到'5'就結束迴圈(代表結束)
    cmp al,'5'
    je bye            ;跳出迴圈
    mov al,[esi+1]    ;若讀到該位置的下一個為'5'在尾端加X(代表字元為奇數，須補滿成兩兩配對)
    cmp al,'5'
    je rearX

    mov al,[esi+1]    ;比較該字與下一個字有無一樣
    cmp al,[esi]
    je addX           ;若一樣就跳至addX
    mov al,[esi]      ;沒有一樣
    mov [edi],al      ;就將兩字輸入至myTwo
```

```
    mov al,[esi+1]
    mov [edi+1],al
    add esi,2         ;下一次往後2字  ABCDEF
    jmp quit          ;               ^ ^(+2)
addX:
    mov al,[esi]      ;一樣
    mov [edi],al      ;將第一個字輸入至myTwo
    mov al,'X'        ;下一個字輸入'X'
    mov [edi+1],al
    add esi,1         ;下次往後接著讀 AABCDE
    jmp quit          ;              ^^(+1)
rearX:
    mov al,[esi]      ;若該字下一個為'5'(代表沒有兩兩配對)
    mov [edi],al
    mov al,'X'
    mov [edi+1],al
    jmp bye
quit:
    mov al,' '
    mov [edi+2],al
    add edi,3
     loop L1
 bye:mov BYTE PTR [edi+2], 0
     popad
     ret
 Twotwo ENDP
```

副程式五：Find1(找到第一個字對應 Key 的位置，得知 row1、col1)
副程式六：Find2(找到第二個字對應 Key 的位置，得知 row2、col2)

```
Find1 PROC
    ;---------------------------第一個字
    mov ecx,25          ;key長度
    cld
    repne scasb         ;若不一樣就往下找
    dec edi             ;edi為找到的位置
    mov w1pos,edi       ;
    mov edx,00000000h
    mov eax,24
    sub eax,ecx         ;該字在key中的第幾個
    div div5
    mov row1,eax        ;得知該字位於哪行哪列
    mov col1,edx
    ret
Find1 ENDP
```

(Find1、Find2 為相同作法)

副程式七：CmpRow(比較兩字的 Row)

```
CmpRow PROC          ;判斷Row是否相同
    mov GO,0         ;若相同 該字位置往右移，若col+1=5，則wpos-4(移至該行最前頭)，否則wpos+1
    mov eax,row1     ;若不相同繼續往下比Col
    cmp eax,row2
    je  Right1       ;row1==row2
    jmp CmpC         ;row1!=row2
Right1:
    inc col1
    mov eax,col1
    cmp eax,5
    je w1Sub4
    inc w1pos
    jmp Right2
w1Sub4:
    sub w1pos,4
    jmp Right2
```

(Right2、w2Sub4 作法與上圖相同)

副程式八：CmpCol(比較兩字的 Col)

```
CmpCol PROC          ;判斷Col是否相同
    mov GO,0         ;若相同 該字位置往下移，若row+1=5，則wpos-20(移至該列最前頭)，否則wpos+5
    mov eax,col1     ;若不相同就找對角位置
    cmp eax,col2
    je  Down1        ;col1==col2
    jmp other        ;col1!=col2
Down1:
    inc row1
    mov eax,row1
    cmp eax,5
    je w1Sub20
    add w1pos,5
    jmp Down2
w1Sub20:
    sub w1pos,20
    jmp Down2
```

(Down2、w2Sub20 作法與上圖相同)

副程式九：Other(不同行不同列)

```
Other PROC
    mov eax,col1        ;比較col誰大(大的往左移(-)，小的往右移(+))
    cmp eax,col2
    ja Col12
    jb Col21
Col12:                  ;col1 > col2
    mov eax,col1
    sub eax,col2
    sub w1pos,eax
    add w2pos,eax
    jmp quit
Col21:                  ;col2 > col1
    mov eax,col2
    sub eax,col1
    sub w2pos,eax
    add w1pos,eax
quit:
    ret
Other ENDP
```

副程式十：InputPF(將處理好字輸到密文中)

```
InputPF PROC
    mov ecx,tmp
    mov edi,w1pos
    mov al,[edi]     ;把找到的值放到myPf
    mov [ebx], al    ;把找到的值放到myPf
    inc ebx

    mov edi,w2pos
    mov al,[edi]     ;把找到的值放到myPf
    mov [ebx], al    ;把找到的值放到myPf
    inc ebx
    mov w1pos,0
    mov w2pos,0
    add esi,1
    ret
InputPF ENDP
```

三、結果

Demo：

```
Please input the plaintext: : I liked bubble sort. I meet you.

Modified plaintext: IL IK ED BU BX BL ES OR TI ME ET YO UX

The ciphertext is: ES KE KC CX IA CS IL NM SK CL KL HN VZ
```