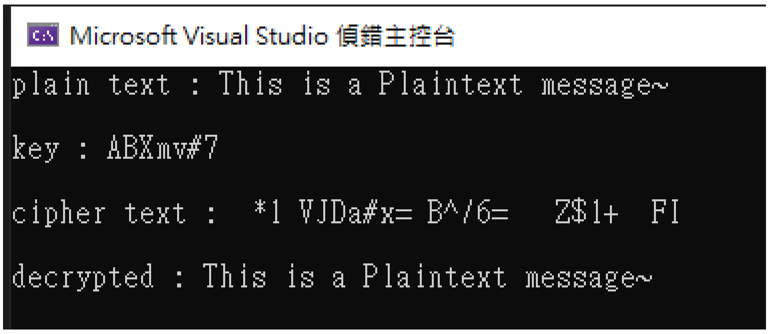
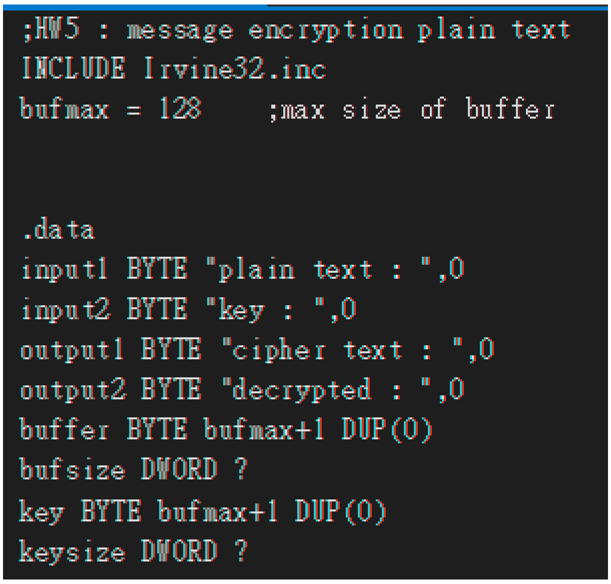
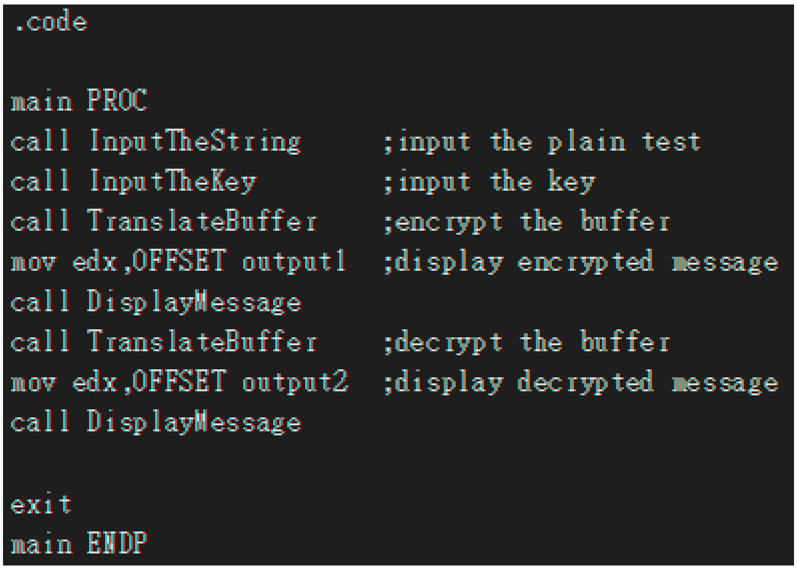


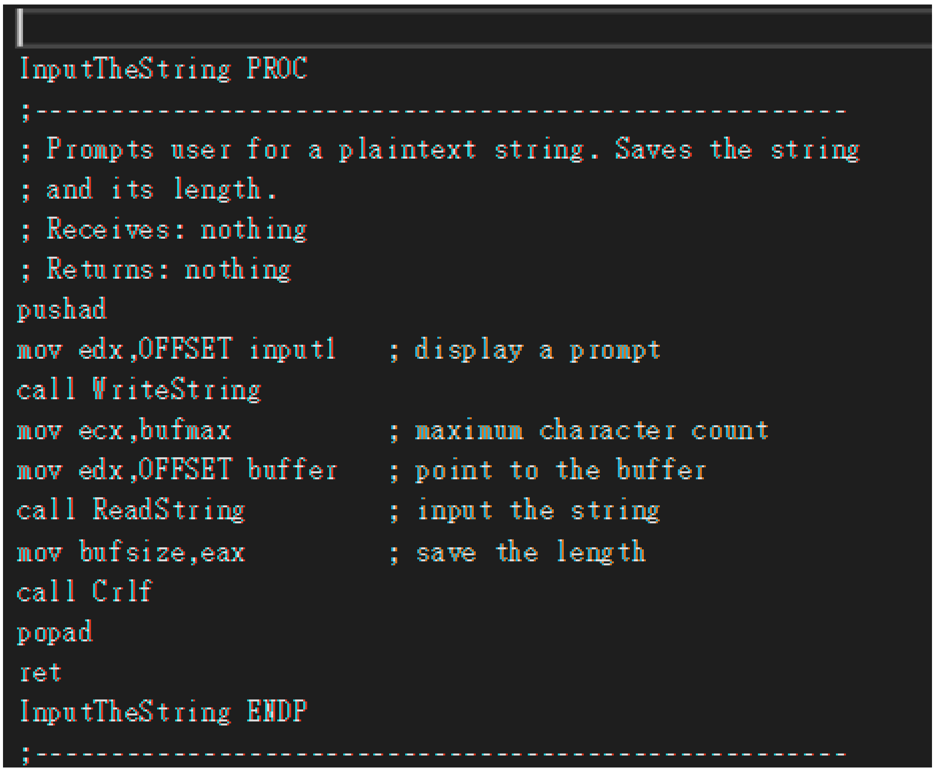
```
plain text : This is a Plaintext message~

key : ABXmv#7

cipher text :  *1 VJDa#x= B^/6=   Z$1+  FI

decrypted : This is a Plaintext message~
```



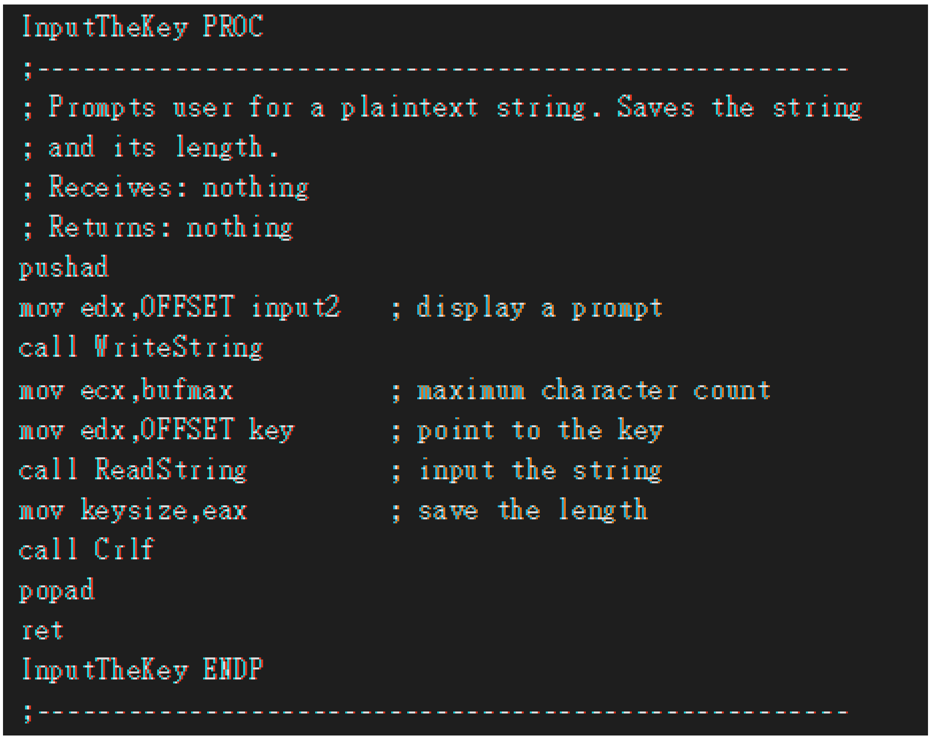
```
;HW5 : message encryption plain text
INCLUDE Irvine32.inc
bufmax = 128     ;max size of buffer


.data
input1 BYTE "plain text : ",0
input2 BYTE "key : ",0
output1 BYTE "cipher text : ",0
output2 BYTE "decrypted : ",0
buffer BYTE bufmax+1 DUP(0)
bufsize DWORD ?
key BYTE bufmax+1 DUP(0)
keysize DWORD ?
```



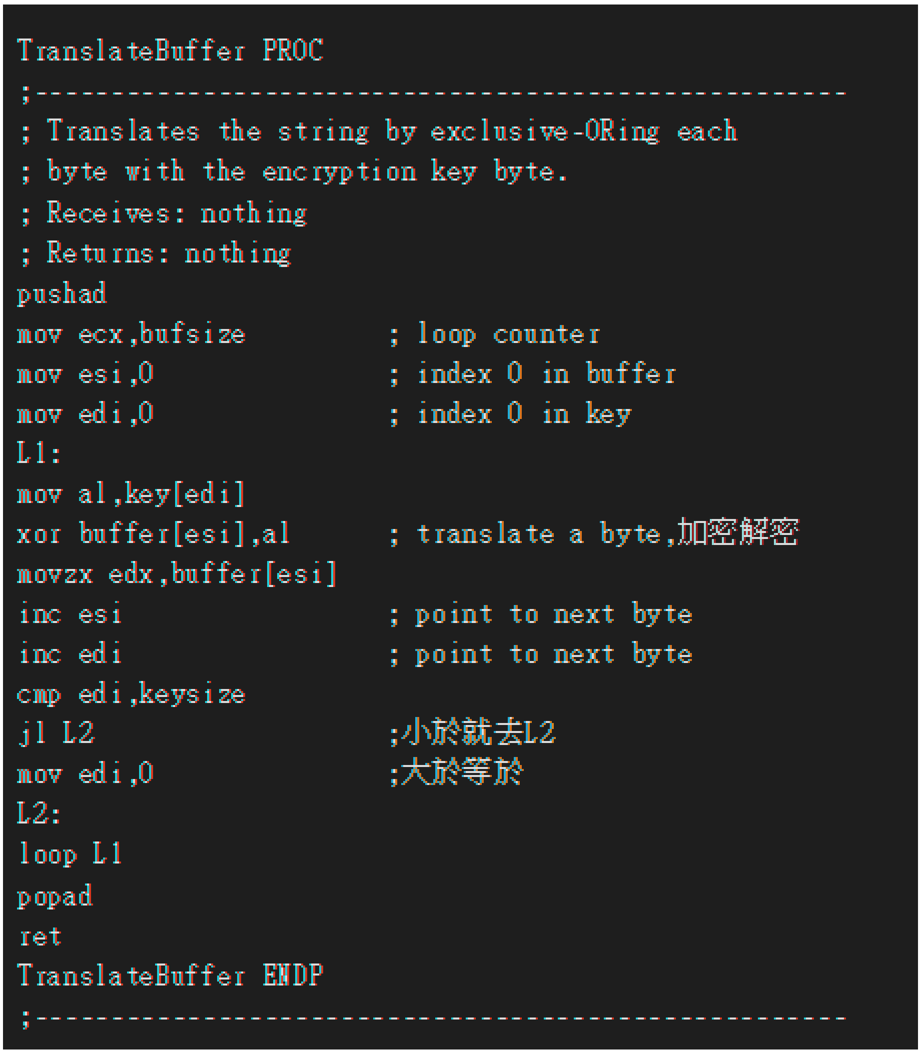
```
.code

main PROC
call InputTheString     ;input the plain test
call InputTheKey        ;input the key
call TranslateBuffer    ;encrypt the buffer
mov edx,OFFSET output1  ;display encrypted message
call DisplayMessage
call TranslateBuffer    ;decrypt the buffer
mov edx,OFFSET output2  ;display decrypted message
call DisplayMessage

exit
main ENDP
```



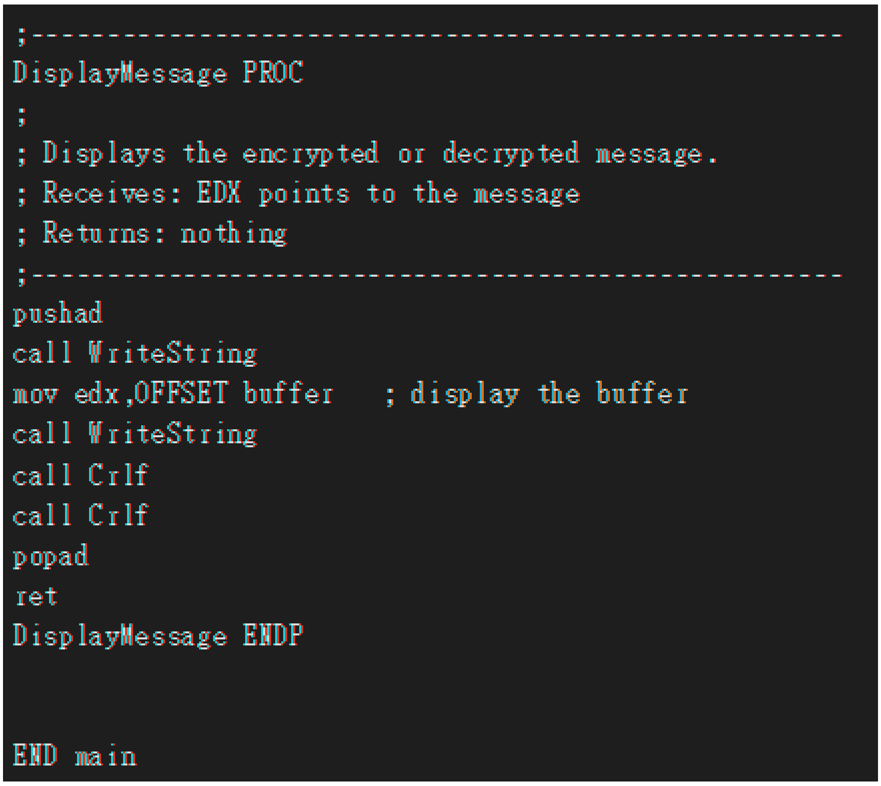
```
InputTheString PROC
;-----------------------------------------------------
; Prompts user for a plaintext string. Saves the string
; and its length.
; Receives: nothing
; Returns: nothing
pushad
mov edx,OFFSET input1    ; display a prompt
call WriteString
mov ecx,bufmax           ; maximum character count
mov edx,OFFSET buffer    ; point to the buffer
call ReadString          ; input the string
mov bufsize,eax          ; save the length
call Crlf
popad
ret
InputTheString ENDP
;-----------------------------------------------------
```



```
InputTheKey PROC
;-----------------------------------------------------
; Prompts user for a plaintext string. Saves the string
; and its length.
; Receives: nothing
; Returns: nothing
pushad
mov edx,OFFSET input2    ; display a prompt
call WriteString
mov ecx,bufmax           ; maximum character count
mov edx,OFFSET key       ; point to the key
call ReadString          ; input the string
mov keysize,eax          ; save the length
call Crlf
popad
ret
InputTheKey ENDP
;-----------------------------------------------------
```



```
TranslateBuffer PROC
;-------------------------------------------------------
; Translates the string by exclusive-ORing each
; byte with the encryption key byte.
; Receives: nothing
; Returns: nothing
pushad
mov ecx,bufsize          ; loop counter
mov esi,0                ; index 0 in buffer
mov edi,0                ; index 0 in key
L1:
mov al,key[edi]
xor buffer[esi],al       ; translate a byte,加密解密
movzx edx,buffer[esi]
inc esi                  ; point to next byte
inc edi                  ; point to next byte
cmp edi,keysize
jl L2                    ;小於就去L2
mov edi,0                ;大於等於
L2:
loop L1
popad
ret
TranslateBuffer ENDP
;-------------------------------------------------------
```



```
;-------------------------------------------------------
DisplayMessage PROC
;
; Displays the encrypted or decrypted message.
; Receives: EDX points to the message
; Returns: nothing
;-------------------------------------------------------
pushad
call WriteString
mov edx,OFFSET buffer    ; display the buffer
call WriteString
call Crlf
call Crlf
popad
ret
DisplayMessage ENDP


END main
```