

```
org 1000h
; variables
string DB "MI_cONTraSEñA.UnlCa"
voc DB "AEIOU"
cant DB 0
```

```
org 3000h
ENCRYPT: PUSH BX ; Guardo el contexto del programa
MOV BX, SP
ADD BX, 4 ; Un push + dir de ret
PUSH AX
PUSH CX
PUSH DX
MOV BX, [BX] ; Recupero la direccion de memoria del caracter
MOV AL, [BX] ; Recupero el valor del caracter
CMP AL, 41H ; Compara el caracter con A
JNZ CONT1
MOV Byte PTR [BX], 40h ; Si es igual lo reemplaza por @
CONT1: CMP AL, 45h ; Compara el caracter con E
JNZ CONT2
MOV Byte PTR [BX], 33h ; Si es igual lo reemplaza por 3
CONT2: CMP AL, 49h ; Compara el caracter con I
JNZ CONT3
MOV Byte PTR [BX], 21h ; Si es igual lo reemplaza por !
CONT3: CMP AL, 4FH ; Compara el caracter con O
JNZ CONT4
MOV Byte PTR [BX], 30h ; Si es igual lo reemplaza por 0
CONT4: CMP AL, 55h ; Compara el caracter con U
JNZ CONT5
MOV Byte PTR [BX], 75h ; Si es igual lo reemplaza por u
CONT5: pop dx ; Recupero el contexto del programa
POP CX
POP AX
POP BX
RET
```

```
ES_VOCAL: PUSH BX ; Guardo el contexto del programa
PUSH CX
PUSH DX
MOV BX, OFFSET voc ; Obtengo la direccion de memoria de las vocales
MOV CH, OFFSET cant - OFFSET voc ; Obtengo la dimension lógica
```

MOV AH, 0h ; Inicializo la variable "boolean" en 0 (falso)
LAZO: CMP CH,0 ; Recorro la string AEIOU, si el caracter es alguna de las vocales mayusculas
le asigna 0FFh sino no hace nada
JZ FIN
DEC CH
CMP AL, [BX]
PUSHF
INC BX
POPF
JNZ LAZO
MOV AH,0FFh ; En caso que sea vocal mayuscula, le asigna a AH FFh (true)

FIN: POP DX ; Recupero el contexto del programa
POP CX
POP BX
RET

org 2000h
; código
MOV BX, OFFSET string ; Obtengo la direccion de memoria de la string a modificar
MOV CL, OFFSET voc - OFFSET string ; Obtengo la longitud de la string
MOV CH, 0 ; Cantidad de modificaciones
LAZO2: CMP CL, 0 ; Comparo que la longitud(dimL) no haya llegado a 0
JZ FINPRO
MOV AL, [BX]
call ES_VOCAL ; Verifico que sea vocal mayúscula
CMP AH, 0FFh ; Comparo el "boolean"
JNZ CONTPRO ; Si es falso continua
PUSH BX ; Si es verdadero encripta el caracter
call ENCRYPT
POP BX
INC CH ; Incrementa la cantidad de modificaciones
CONTPRO: DEC CL ; Decrementa la dimL
INC BX ; Avanza de caracter en la string
JMP LAZO2

FINPRO: MOV cant, CH ; Guardo en memoria la cantidad de modificaciones
hlt
end