***COAL LAB #12***

***18F-0358\_Mehdi Raza***

***Q#1:-***

include Irvine32.inc

.data

op1 BYTE 34h,12h,98h,74h,06h,0A4h,0B2h,0A2h

op2 BYTE 02h,45h,23h,00h,00h,87h,10h,80h

opp byte 8 dup(?)

.code

main proc

mov dl,0h

mov edi, offset op2

mov ecx,lengthof op1

mov ebx,offset opp

mov esi,offset op1

l1:

mov al,[esi]

adc al,[edi]

mov[ebx+ecx],al

inc edi

inc esi

loop l1

mov ebx,type opp

mov esi ,offset opp

mov ecx,lengthof opp

call DumpMem

exit

main endp

end main

***Q#2:-***

Encryption using rotate operation

INCLUDE Irvine32.inc  
.data  
  
key BYTE -2, 4, 1, 0, -3, 5, 2, -4, -4, 6  
keySize = $ - key  
plainText BYTE "This plaintext message will be encrypted.",0  
  
.code  
main PROC  
  
   call Clrscr  
  
L1:   mov edx, OFFSET plainText  
   call WriteString  
   mov esi, 0  
   movsx eax, key[esi]  
   call WriteInt  
   mov ecx, keySize  
   call Encode  
   jnz L1  
  
   mov edx, OFFSET plainText  
   call WriteString  
   call Crlf  
  
main ENDP  
  
Encode PROC  
  
L1:   push ecx  
   cmp BYTE PTR[esi],0  
   je L4  
  
   mov cl, [esi]  
   cmp cl, 0  
   jge L2  
   neg cl  
   rol BYTE PTR[esi], cl  
   jmp L3  
  
L2:   ror BYTE PTR[esi], cl  
  
L3:   inc esi  
   inc edi  
   pop ecx  
   loop L1  
   or eax, 1  
   jmp L5  
  
L4:   pop ecx  
  
L5:   ret  
  
Encode ENDP  
  
END main

***Q#3:-***

Finding relative prime numbers

include Irvine32.inc

.data

no1 dw 0120

no2 dw 0090

gcd dw 0h

.code

main proc

mov ax,no1

mov bx,no2

w:

cmp ax,bx

je l1

jb v1

l2:

mov dx,0

div bx

cmp dx,0

je l1

mov ax,dx

jmp w

v1:

xchg ax,bx

jmp l2

l1:

mov gcd,04h

mov cl,04h

ll2:

rol bx,cl

mov dl,0fh

and dl,0fh

cmp dl,09

jbe l4

add dl,07

l4:

add dl,30h

mov ah,02

dec ch

jnz ll2

mov ax,bx

exit

main endp

end main