*Program 12.1:*

include Irvine32.inc

.data

Op1 byte 34h,12h,98h,74h,06h,0A4h,0B2h,0A2h

Op2 byte 02h,45h,23h,00h,00h,87h,10h,80h

Op3 byte lengthof op1 dup(?)

.code

main proc

mov esi offset op1

mov ebx offset op2

mov edi offset op3

mov ecx, lengthof op1

call extended\_add

call display\_sum

call readchar

exit

main endp

extended\_add proc

clc

L1:

Mov al,[esi]

Adc al,[ebx]

Mov [edi],al

Inc esi

Inc edi

Inc ebx

Loop L1

Jnc rest

Mov al,’1’

Call writechar

restL

ret

extended\_added endp

Display\_sum proc

Mov eax,DWORD PTR op3+4

Call writehex

Ret

Display\_sum endp

end main

*Program 12.2:*

include Irvine32.inc

.data

Key byte -2,4,1,0,-3,5,2,-4,-4,6

Var1 byte “ENTER THE STRING”,0

Var2 byte 11 DUP(?)

.code

Main proc

Call stringinput

Call encrypt

Call printstring

Call readchar

Exit

Main endp

Stringinput proc

Mov edx,offset var1

Call writestring

Mov edx,offset var2

Mov ecx,10

Call readstring

Ret

Stringinput endp

Printstring proc

Mov edx,offset var2

Call writestring

Ret

Printstring endp

Encrypt proc

Mov esi,offset key

Mov ecx,lengthof key

Mov edi,offset var2

L1:

Mov eax,ecx

Mov cl,[esi]

Mov bl,[edi]

Cmp cl,0

Jb rotateleft

Rotate right:

Ror bl,cl

Jmp next

Rotate left:

Not cl

Rol bl,cl

Next:

Mov [edi],bl

Inc esi

Inc edi

Mov ecx,eax

Loop L1

Ret

Encrypt endp

end main

*Program 12.3:*

Include32.inc

.data

Val1 byte?

Val2 byte?

Val3 byte?

Pmp1 byte “ENTER FIRST NUMBER :”,0

Pmp2 byte “ENTER SECOND NUMBER :”,0

Pmp3 byte “GCD :”,0

.code

Main proc

Call dec\_in

Call GCD\_ab

Call readchar

Exit

Main endp

DEC\_in proc

Input 1:

Mov edx,offset pmp1

Call writestring

Call readint

Cmp al,0

Jl input1

Cmp al,99

Jg input1

Mov var1,al

Input2:

Mov edx,offset pmp2

Call writestring

Call reading

Cmp al,0

Jl input2

Cmp al,99

Jg input2

Mov var2,al

Ret

DEC\_in endp

Gcd\_ab proc

Movzx eax,var1

Movzx ebx,var2

Division:

Cmp ax,bx

Je found

Ja next

XCHG ax,bx

Next:

Div bl

Cmp ah,0

Je found

XCHG bl,ah

Movzx ax,ah

Jmp next

Found:

Mov var3

Ret

Gcd\_ab endp

DEC\_out proc

Mov edx,offset pmp3

Call Writestring

Movzx eax,var3

Call writedec

Ret

Dec\_out endp

End main