*Program 13.1:*

Include Irvine32.inc

.data

string1 BYTE "Enter String for Evaluation ", 0

string2 BYTE "Vowels Found are ", 0

string3 BYTE "Consonants Found are ", 0

string4 BYTE "Number of Vowels are ",0

string5 BYTE "Number of Consonants are ",0

vowels BYTE "aeiou"

string\_In BYTE 26 DUP(? )

vowels\_count BYTE sizeof vowels

RVowels BYTE lengthof string\_In DUP(? )

RConsonants BYTE lengthof string\_In DUP(? )

count\_vowels BYTE 0

count\_consonants BYTE 0

.code

main proc

call STR\_IN

mov esi, OFFSET string\_In

mov ebp, OFFSET RVowels

mov ebx, OFFSET RConsonants

call SEARCH

call DIS\_STR

call crlf

call waitmsg

exit

main endp

STR\_IN PROC

mov ecx,lengthof string1

mov edx, OFFSET string1

call WriteString

call CRLF

mov edx, OFFSET string\_In

mov ecx, lengthof string\_In

call ReadString

RET

STR\_IN ENDP

SEARCH PROC

cld

mov ecx, eax

L1 :

mov edx, ecx

mov edi, OFFSET vowels

lodsb

movzx ecx, vowels\_count

repne scasb

je LABEL\_2

mov[ebx], al

INC ebx

inc count\_consonants

jmp LABEL\_1

LABEL\_2 :

mov[ebp], al

INC ebp

inc count\_vowels

LABEL\_1 :

mov ecx, edx

loop L1

RET

SEARCH ENDP

DIS\_STR PROC

mov edx, OFFSET string2

call WriteString

mov edx, OFFSET RVowels

call WriteString

call crlf

mov edx, OFFSET string3

call WriteString

mov edx, OFFSET RConsonants

call WriteString

call CRLF

call CRLF

mov ecx,lengthof string4

mov edx,offset string4

call writestring

call CRLF

mov al,count\_vowels

call writehex

call CRLF

call CRLF

mov ecx,lengthof string5

mov edx,offset string5

call writestring

call CRLF

mov al,count\_consonants

call writehex

RET

DIS\_STR ENDP

end main

*Program 13.2:*

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

