

8086 Assembly language programs

SECTION A:

i) To print the string "Hello Pokhara University"

```
.MODEL SMALL
.STACK
.DATA
    STRING DB 10,13, 'HELLO Pokhara University $' ; declaring STRING
.CODE
MAIN PROC
    MOV AX, @DATA ; main procedure
    MOV DS, AX ; initialize the data segment
    LEA DX, STRING ; loading the effective address
    MOV AH, 09H ; for string display
    INT 21H ; dos interrupt function
    MOV AX, 4C00H+退出 ; all 退出 AL
    INT 21H ; 等待键入字符
MAIN ENDP ; end procedure
END MAIN ; end program
```

ii) To print the string "Hello Pokhara University, GOODMORNING STUDENTS" IN
3 DIFFERENT LINES

```
.MODEL SMALL
.STACK
.DATA
    STRING1 DB 'HELLO Pokhara University $'
    STRING2 DB 10,13, 'GOOD MORNING $'
    STRING3 DB 10,13, 'STUDENTS $' ; declaring STRING
.CODE
MAIN PROC ; main procedure
    MOV AX, @DATA ; initialize the data segment
    MOV DS, AX
```

```

LEA DX, STRING1 ; loading the effective address
MOV AH, 09H ; for string display

INT 21H

LEA DX, STRING2

MOV AH, 09H

INT 21H

LEA DX, STRING3

MOV AH, 09H ;dos interrupt function

INT 21H ; end request

MOV AX, 4C00H

INT 21H ; end procedure

MAIN ENDP ; end program

END MAIN

```

I)iii) to read the string character by character and display the same string in screen.

```

.model small
.stack 100h
.data
msg db 13,10,'enter the string:$'
msg2 db 13,10,'you entered:$'
str1 db 255 dup()

```

Note

Int 21h functions.
 02h = Display character
 03h = Display string
 0Ah = Input from keyboard
 3Fh = Input from keyboard
 40h = Display string

Int 20h functions.

02h = set cursor
 06h = scroll screen.

Eg. to set row 08 & column 15

mov ah, 02h
 mov dh, 08
 mov dl, 15
 int 10h } → OR { mov ah, 09h
 mov dx, 080f h
 int 10h vom

.code

main proc

mov ax,@data

mov ds,ax

lea dx,msg

mov ah,09h ;display prompt msg

int 21h

lea si,str1

mov ah,01h ;input string

read:

int 21h

mov [si],al

inc si ;read input string character by character

cmp al,0dh

jne read

display:

mov al,'\$'

mov [si],al

lea dx,msg2

mov ah,09h

int 21h

lea dx,str1

mov ah,09h

int 21h

mov ah,4ch

int 21h

main endp

end main

2) To add two 8-bit Numbers

.MODEL SMALL

.STACK

.DATA

D1 DB 23

;first 8 bit data

D2 DB 13

MSG DB 'SUM =\$' ; second 8 bit data

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AX,0

MOV AL,DI

ADD AL,D2

AAM ;AAM converts binary to unpacked BCD

ADD AX,3030H ; ax is added to 3030h to obtain ascii value

PUSH AX

LEA DX,MSG ;display message

MOV AH,09H

INT 21H

POP AX

MOV DL,AH

MOV DH,AL

MOV AH,02H ;print character on screen

INT 21H

```
MOV DL,DH
MOV AH,02H           ;print character on screen
INT 21H
MOV AX,4C00H
INT 21H
MAIN ENDP
END MAIN
```

3) To subtract two 8 bit Numbers

```
.MODEL SMALL
.STACK
.DATA
    D1 DB 23H           ;first 8 bit data
    D2 DB 13H           ; second 8 bit data
.CODE
    MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        MOV AL,D1
        SUB AL,D2
        MOV AH,02           ; to print difference in screen
        INT 21H
        MOV AX,4C00H
        INT 21H
    MAIN ENDP
    END MAIN
```

4) To find sum of two numbers as input from user and print sum in the screen

```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1 DB 'ENTER FIRST NUMBER: $'
    MSG2 DB 10,13,'ENTER SECOND NUMBER: $'
    MSG3 DB 10,13,'THE SUM IS: $'
.CODE
    MAIN PROC
        MOV AX, @DATA
        MOV DS, AX
        LEA DX, MSG1
        MOV AH, 09          ; to display message1
        INT 21H
        MOV AH, 01          ;input first number
        INT 21H
        MOV BL, AL           ;copy that number to bl
        AND BL, 0FH         ;converting ascii code to hexa code

        LEA DX, MSG2          ;to display message2
        MOV AH, 09
        INT 21H
        MOV AH, 01          ;input second number
        INT 21H

        AND AL, 0FH
        ADD BL, AL
        LEA DX, MSG3
        MOV AH, 09
        INT 21H
        MOV DL, BL           ;copying sum to register dl

OR DL,30H             ;converting HEX TO ASCII
        MOV AH, 02          ;to print sum on screen
        INT 21H
        MOV AX, 4C00H
        INT 21H
```

LEA BX, DATA3
MOV CX, 03
MOV AX,
Back: ADC AX,
mov

MAIN ENDP
END MAIN

5) To find the sum of two sixteen bit data

```
.MODEL SMALL
.STACK
.DATA
    DATA1 DW 0231H      ;first word
    DATA2 DW 0345H      ;second word
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    MOV BX, DATA1
    ADD BX, DATA2

    MOV AH, 4CH
    INT 21H
MAIN END
END MAIN
```

6) TO add two multiword numbers

```
.MODEL SMALL
.STACK
.DATA
    DATA1 DW 1D54H, 8F99H, 63CEH      ;allocate 3 words
    DATA2 DW 3F73H, 4FA2H, 3B8DH
    DATA3 DW 3 DUP(?)                ; save result

.CODE
MAIN PROC FAR
    MOV AX, @DATA
    MOV DS, AX
    CLC                      ;clear carry
    LEA SI, DATA1
    LEA DI, DATA2
```

```
LEA BX, DATA3  
MOV CX, 03
```

```
Back: MOV AX, [SI]  
      ADC AX, [DI]  
      MOV [BX], AX  
      ADD SI, 2  
      ADD DI, 2  
      ADD BX, 2  
      DEC CX  
      JNZ Back ;when cx is not zero then goto back  
MOV AH, 4CH  
INT 21H  
MAIN ENDP  
END MAIN
```

7) TO add two sixteen digit packed BCD numbers

```
.MODEL SMALL  
.STACK  
.DATA  
  SOURCE1 DB 01H, 08H, 99H, 42H, 11H, 46H, 32H, 55H  
  SOURCE2 DW 55H, 44H, 66H, 41H, 83H, 21H, 41H, 84H  
  RESULT DW 8 DUP(?)  
.CODE  
MAIN PROC  
MOV AX, @DATA  
MOV DS, AX  
MOV CX, 08 ; initialize counter with 8  
LEA SI, SOURCE1  
LEA DI, SOURCE2  
LEA BX, RESULT  
CLC ;Clear carry flag  
  
BEG:  
  MOV AL, [SI]  
  ADC AL, [SI]  
  DAA ;Decimal adjust accumulator  
  MOV [BX], AL
```

```
INC SI  
INC DI  
INC BX  
  
LOOP BEG      ; IF CX not zero go to label Beg  
MOV AX,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

8) To display the given series as A B C D.....Z

```
.MODEL SMALL  
.STACK  
.DATA  
.CODE  
MAIN PROC  
MOV CX,1AH      ; LOAD CX with count of 26 in hex  
MOV DL,41H      ; LOAD dl with ascii value of A  
AGAIN:  
    MOV AH,02      ; PRINT character On screen  
    INT 21H  
  
    INC DL  
    LOOP AGAIN     ;LOOP 26 times  
  
    MOV AH,4CH  
    INT 21H  
MAIN ENDP  
END MAIN
```

9) TO print a,b,c,d,e.....v,z in single line

```
.MODEL SMALL  
.DATA  
    a DB 41H  
    b DB 61H  
.CODE  
MAIN PROC
```

```
MOV AX, @DATA  
MOV DS, AX
```

```
LEA SI, a  
MOV CL, 26 ;COUNTER of 26 alphabets  
MOV DL, [SI] ;LOADING index to dl
```

UP:

```
MOV AH, 02H  
INT 21H  
INC DL
```

```
DEC CL
```

```
JNZ UP  
MOV AH, 4CH  
INT 21H  
MAIN ENDP  
END MAIN
```

10) To print from 0,1.....9

```
.MODEL SMALL  
.STACK 100H  
.DATA  
PROMPT DB 'THE COUNTING FROM 0 TO 9 IS $'  
.CODE  
  
MAIN PROC  
MOV AX, @DATA  
MOV DS, AX  
LEA DX, PROMPT  
MOV AH, 09  
INT 21H  
MOV CX, 10 ;initialize counter for 10 numbers  
MOV AH, 02  
MOV DL, 48 ;SET DL WITH 0
```

LOOP:

```
INT 21H  
INC DL  
DEC CX  
JNZ LOOP  
MOV AH,4CH  
INT 21H
```

```
MAIN ENDP  
END MAIN
```

SECTION B:

11) To add a list of 10 data

```
TITLE ADDING A LIST OF 10 DATA  
DOSSEG  
.MODEL SMALL  
.STACK 100H  
.DATA  
ARRAY1 DB 00,01, 02,03,04,05,06,07,08,09  
TOTAL DW ?  
.CODE  
MAIN PROC  
  
MOV AX, @DATA  
MOV DS, AX  
  
MOV CX, 10 ; COUNTER OF 10 data  
LEA SI, ARRAY1  
MOV DX,0000 ;SUM=0  
NEXT:  
    ADD DL,[SI]  
    ADC DH, 0  
    INC SI  
LOOP NEXT ;LOOP until cx=0  
MOV TOTAL, DX  
MOV AX,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

```
MOV CX,10  
SUB AX, AX  
UP:  
    ADD AX, CX  
    DEC CX  
    JNZ UP
```

```
MOV AH,4CH
```

```
INT 21H  
MAIN ENDP  
END MAIN
```

14) A program to print sum of even numbers from 0 to 10

```
.MODEL SMALL  
.STACK 100H  
.DATA  
.CODE  
MAIN PROC  
MOV AX,0  
MOV DS,AX  
MOV CX,10  
SUB AX, AX      ; TO STORE result(AX=0)  
UP:  
    ADD AX, CX  
    DEC CX  
    DEC CX  
    JNZ UP
```

```
MOV AH,4CH
```

```
INT 21H  
MAIN ENDP  
END MAIN
```

15) To find the average of two numbers

12) TO copy a block of data from one array to another

```
TITLE COPY BLOCK OF DATA
DOSSEG
.MODEL SMALL
.STACK 100H
.DATA
    ARRAY1 DB 00,01,02,03,04,05,06,07,08,09
    ARRAY2 DB 10 DUP(?)
.CODE

MAIN PROC
MOV AX, @DATA
MOV DS,AX
MOV CX,0AH
LEA SI, ARRAY1
LEA DI, ARRAY2
NEXT:
    MOV BL,[SI]      ;COPY TO bl from array1
    MOV [DI], BL      ; transfer to other array
    INC SI
    INC DI
LOOP NEXT
MOV AX,4C00H
INT 21H
MAIN ENDP
END MAIN
```

13)a program to add natural numbers from 1 to 10

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
```

```
.MODEL SMALL  
.STACK 100H  
.DATA  
  
DATA1 DB 63H  
DATA2 DB 2EH  
AVG DB ?  
.CODE
```

```
MAIN PROC  
MOV AX,@DATA  
MOV DS, AX  
MOV AL,DATA1  
ADD AL,DATA2  
ADC AH,00H      ;PUT CARRY in ah  
MOV BL,02H      ;DIVIDE sum by 2  
DIV BL  
MOV AVG, AL  
MAIN ENDP  
END MAIN
```

16)A program to multiply two numbers inputted by the user and display result in screen

```
.MODEL SMALL  
.STACK  
.DATA  
.CODE  
  
MAIN PROC  
MOV AH,01H      ;INPUT first number  
INT 21H  
MOV BL,AL  
AND BL,0FH  
MOV AH,01H      ;COPY first number to bl  
;CONVERT asciicode to hex code  
INT 21H          ;TAKE second input number  
  
AND AL,0FH      ;CONVER second asciicode num to hex
```

```
MUL BL           ;MULTIPLY two numbers
MOV DL,AL
OR DL,30H         ;CONVERT hex product to its ascii equivalent
MOV AH,02H
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

17) A program to find the highest (maximum) among the 5 datas and store the result in dl register

```
MODEL SMALL
.STACK
.DATA
    SCORES DB 51, 44, 99, 93,80      ;13H,2CH,63H,58H,50H
.CODE
    MAIN PROC
        MOV AX,@DATA
        MOV DS, AX
        LEA BX, SCORES

        MOV CX, 5          ;COUNTER FOR 5 data
        SUB AL,AL          ;al=0,ahighest value

        AGAIN:
            CMP AL,[BX]    ;compare next data to highest
            JA NEXT          ;jump if AL Still highest
            MOV AL,[BX]        ;Else al holds new highest

        NEXT:
            INC BX          ;POINT to next score
            LOOP AGAIN       ;CONTINUE comparing
            MOV DL,AL

        MOV AX,4C00H
```

```
INT 21H  
MAIN ENDP  
END MAIN
```

18) A PROGRAM TO COUNT NO. OF 1'S IN A BYTE AND WRITES RESULT INTO BL

```
.MODEL SMALL  
.STACK  
.DATA  
    DATA1 DB 97 ; 61H  
.CODE  
MAIN PROC  
MOV AX,@DATA  
MOV DS, AX  
  
SUB BL,BL      ;Clear bl to keep no. of 1s  
MOV DL,8       ;8 TIMES  
  
MOV AL,DATA1  
AGAIN:  
  
ROL AL,1       ;ROTATE it once  
JNC NEXT       ;CHECK if 1  
INC BL         ;IF CF=1 THEN add one to count  
  
NEXT:  
DEC DL  
JNZ AGAIN  
  
MOV AX,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

19) A program to reverse a bit pattern

```
.MODEL SMALL
```

```
.STACK
.DATA

.CODE
MAIN PROC
MOV AX,00ABH      ;LOAD ax with required data
MOV CX,08          ;LOAD count of 8
AGAIN:
RCL AL,1           ;rotate left through carry content of reg so msb shifted to
                   ;carry
RCR DL,1           ;SHIFT The contents of carry flag to msb of reG dl
LOOP AGAIN
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

20) A program to copy the string from one location in memory to other

```
MODEL SMALL
.STACK
.DATA
    STRING1 DB 'GOOD MORNING $'
    STRING2 DB ' $'
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX

MOV ES,AX
MOV CX,12
LEA SI,STRING1
LEA DI,STRING2
REP MOVSB    ; REPEAT the movement of characters in string from
;source to destination till count become 0
MOV AH,4CH
INT 21H
MAIN ENDP
```

END MAIN

SECTION C:

21)A PROGRAM TO CONVERT BINARY(HEX) TO ASCII(DECIMAL)
CONVERSION

```
.MODEL SMALL
.STACK
.DATA
    BINNUM DW 34DH
    ASCNUM DB DUP('0')
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
MOV BX,10
MOV SI,5
DEC SI
MOV AX,BINNUM
BACK:
    SUB DX,DX
    DIV BX
    OR DL,30H
    MOV [SI],DL
    DEC SI
    CMP AX,0
    JA BACK
    INT 20H
    MOV AX,4C00H
    INT 21H
MAIN ENDP
END MAIN
```

EXPLANATION:

$$34DH = 3 * 256 + 4 * 16 + 13 * 1 = 845$$

$34\text{DH}/\text{A} = 84$ remainder 5

$84\text{H}/\text{A} = 8$ remainder 4

$8 < \text{A}$ = THE process stops

Taking remainder in reverse order gives 8 4 5 decimal value

22) To read a string and convert to upper case and display

```
TITLE
.MODEL SMALL
.STACK
.DATA
    MAX_CHAR DB 20
    ACT_CHAR DB ?
    STR DB 20 DUP(?)
                                ;to store count of characters

.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX
LEA DX, MAX_CHAR
MOV AH, 0AH
INT 21H

MOV AH, 02H
MOV DL, 0DH
INT 21H

MOV DL, 0AH
INT 21H
MOV CL, ACT_CHAR
MOV CH, 00H
MOV BX, 0000H

UP:
MOV DL, STR[BX]
```

```
CMP DL, 97D  
JC DOWN  
CMP DL, 123D  
JNC DOWN  
ADD DL,-32
```

DOWN:

```
MOV AH, 02  
INT 21H  
INC BX  
LOOP UP
```

```
MOV AX,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

23) To read a string and display each word in center of clear screen with blue background and cyan foreword

```
TITLE  
DOSSEG  
.MODEL SMALL  
.STACK  
.DATA  
MAXCHAR DB 100  
ACTCHAR DB ?  
WDCOUNT DB 0  
.CODE  
MAIN PROC  
MOV AX,@DATA  
MOV DS, AX  
LEA DX,MAXCHAR  
MOV AH, 0AH ;INPUT STRING  
INT 21H  
  
MOV AH,0  
MOV AL,0  
INT 10H
```

```
MOV DL, 40H  
MOV DH, WDCOUNT  
MOV AH,02  
INT 10H
```

```
MOV CL, ACTCHAR  
MOV CH,00H  
MOV BX,0  
UP:  
    MOV DL, STR[BX]  
    CMP DL,32  
    JNZ DOWN  
    INC WDCOUNT  
    MOV AH,02  
    MOV DL,0DH  
    INT 21H  
    MOV DL, 0AH  
    INT 21H  
DOWN:  
    MOV AX,4C00H  
    INT 21H  
MAIN ENDP  
END MAIN
```

24)To read the string count the vowels and display them in a clear screen with reverse attribute

```
TITLE  
.MODEL SMALL  
.STACK  
.DATA  
    MAXCHAR DB 60  
    ACTCHAR DB ?  
    VOWELNO DB 0  
    NEWLINE DB 0DH,0AH,"$"  
    CHECKSTR DB 65,69,73,79,85,97,101,105,111,117  
.CODE
```

```
MAINPROC
MOV AX,@DATA
MOV DS, AX
LEA DX, MAXCHAR
MOV AH,0AH
INT 21H
CALL 21H
CALL NEXTLINE
MOV CL,ACTCHAR
MOV CH,0
MOV BX,0
A1:
PUSH CX
MOV DL, STR[BX]
MOV SI,0
MOV CX,10
A2:
INC VOWELNO
MOV AH,02H
INT 21H
A4:
INC BX
POP CX
LOOP A1
CALL VOWELCOUNTDISPLAY
MOV AX,4C00H
INT 21H
MAIN ENDP
```

```
NEXTLINE PROC
LEA DX,NEWLINE
MOV AH,09H
INT 21H
RET
NEXTLINE ENDP
```

```
VOWELCOUNTDISPLAY PROC
MOV CX,0
MOV AL,VOWELNO
MOV AH,0
```

```
MOV BX,0
LABEL1:
    MOV DX,0
    DIV CX
    ADD DX,30H
    PUSH DX
    INC BX
    CMP AX,0
    JC LABEL1
    MOV AH,02
    MOV CX,BX
    DISP:
        POP DX
        INT 21H
    LOOP DISP
VOWELCOUNTDISPLAY ENDP
END MAIN
```

25) TO DISPLAY ASCII character from 32 to 127

```
.MODEL SMALL
.STACK
.DATA

.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
MOV AH,02H
MOV DL,32
LABEL1:
    INC DL
    INT 21H
    CMP DL,127
    JB LABEL1

MOV AX,4C00H
INT 21H
```

MAIN ENDP
END MAIN

26) To display given TEXT in separate lines

```
.MODEL SMALL
.STACK
.DATA
    MAXCHAR DB 100
    ACTCHAR DB ?
    STR DB 100 DUP(?)
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
LEA DX,MAXCHAR
MOV AH,0AH
INT 21H

MOV AH,02
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H
MOV CL,ACTCHAR
MOV CH,00H
11:
MOV DL,STR[BX]
CMP DL,32
JNA JUM
MOV AH,02
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H
JMP 12
JUM:
MOV AH,02H
```

```
INT 21H  
12:  
INC BX  
LOOP 11  
MOV AX,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

27) TO reverse a string inputted from the user

```
.MODEL SMALL  
.DATA  
    STR1 DB 5 DUP(" "), '$'  
    STR2 DB 5 DUP(" "), '$'  
    NL DB 0DH,0AH, '$'  
.CODE  
MAIN PROC  
  
MOV AX,@DATA  
MOV DS,AX  
LEA SI, STR1  
LEA DI, STR2  
MOV AH,01H  
AGAIN:  
  
INT 21H  
CMP AL,0DH  
JE BAK  
MOV [SI],AL  
INC SI  
JMP AGAIN  
BAK:  
    MOV AL,'$'  
    MOV [SI],AL  
    MOV AH,09H  
    LEA DX,NL
```

```
INT 21H
MOV CX, LENGTH STR1
ADD CX,I
DOTHIS:
    DEC SI
    MOV AL,[SI]
    MOV [DI],AL
    INC DI
LOOP DOTHIS
MOV AL,'$'
MOV [DI],AL
MOV AH,09H
LEA DX,NL
INT 21H
LEA DX,STR2
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

28) TO accepts the number from the user through keyboard calculates its factorial and prints on screen

```
.MODEL SMALL
.STACK
.DATA
    MSG1 DB 'ENTER THE NUMBER :$'
    MSG2 DB 10,13, ' THE FACTORIAL IS:$'
.CODE
```

```
MAIN PROC
MOV AX,@DATA
MOV DS,AX
LEA DX, MSG1
MOV AH,09
INT 21H
MOV AH,01
INT 21H
```

```
XOR AH,AH  
AND AL,000FH  
MOV CX,AX  
MOV AX,01H
```

AGAIN:

```
MUL CX  
LOOP AGAIN  
MOV CX,AX  
LEA DX,MSG2  
MOV AH,09  
INT 21H  
MOV DX,CX  
OR DX,30H  
MOV AH,02  
INT 21H  
MOV AH,02  
INT 21H  
MOV AH,4CH  
INT 21H  
MAIN ENDP  
END MAIN
```

29) TO sort array of five numbers using bubble sort technique

```
.MODEL SMALL  
.STACK  
.DATA  
    LIST DW 22H, 18H, 20H, 21H, 23H  
.CODE  
MAIN PROC  
  
MOV AX,@DATA  
MOV DS,AX  
MOV DX,05H  
AGAIN:  
    MOV AX,[SI]
```

```
CMP AX,[SI+2]
JG PR1
XCHG [SI+2],AX
XCHG[SI],AX
PR1:
    ADD SI,02
LOOP AGAIN:
MOV BX, AX
DEC DX
JNZ AGAIN
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

30) TO FIND THE FIBONACCI SERIES

```
MODEL SMALL
STACK
.DATA
.CODE
    MAIN PROC
MOV SI,0006H
MOV BL,30H
MOV CL,31H
MOV DL,BL
MOV AH,02H
INT 21H
MOV DL,0AH
MOV AH,02H
INT 21H
JUMP:
    MOV AL,BL
    ADD AL,CL
```

```
AAA
OR AL,30H
MOV CL,BL
MOV BL,AL
```

```
MOV DL,AL  
MOV AH,02H  
INT 21H  
MOV DL,0AH  
MOV AH,02H  
INT 21H  
MOV DL,0DH  
MOV AH,02H  
INT 21H  
DEC SI  
JNZ JUMP  
MOV AH,4C00H  
INT 21H  
MAIN ENDP  
END MAIN
```

8086 MASM