ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DB 04H

NUM2 DB 03H

NUM3 DB 04H

SUM DB 01H DUP(?)

CARRY DB 01H DUP(?)

DATA ENDS

CODE SEGMENT

START:MOV AX,DATA

MOV DS,AX

MOV CL,00

MOV AL,NUM1

MOV BL,NUM2

ADD AL,BL

JNC L1

INC CL

L1:MOV BL,NUM3

ADD AL,BL

JNC L2

INC CL

L2:MOV SUM,AL

MOV CARRY,CL

MOV AX,4C00H

INT 21H

CODE ENDS

END START



-------------------------------------------

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DW 0A427H

NUM2 DW 0FE67H

NUM3 DW 06A27H

SUM DW 01h DUP(?)

CARRY DW 01h DUP(?)

DATA ENDS

CODE SEGMENT

START:MOV AX,DATA

MOV DS,AX

mov CX,0000h

MOV AX,NUM1

MOV BX,NUM2

ADD AX,BX

JNC L1

INC CX

L1:MOV BX,NUM3

ADD AX,BX

JNC L2

INC CX

L2:MOV SUM,AX

MOV CARRY,CX

MOV AX,4C00H

INT 21H

CODE ENDS

END START



-------------------------------------------

assume cs:code,ds:data

data segment

num1 dw 0abcfh

num2 dw 00c4h

prol dw 01h dup(?)

proh dw 01h dup(?)

data ends

code segment

start:mov ax,data

mov ds,ax

mov ax,num1

mov bx,num2

mul bx

mov prol,ax

mov proh,dx

mov ax,4c00h

int 21h

code ends

end start



-------------------------------------------

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

div1 DW 7004H

div2 DW 01c1H

dvr dw 020ch

quo1 dw 01h dup(?)

quo2 dw 01h dup(?)

DATA ENDS

CODE SEGMENT

START:MOV AX,DATA

MOV DS,AX

MOV AX,div1

mov DX,div2

mov bx,dvr

div BX

mov quo1,ax

mov quo2,dx

MOV AX,4C00H

INT 21H

CODE ENDS

END START



---------------------------------------------

assume cs:code,ds:data

data segment

arr dw 25h,43h,0aah,12h,10h

len dw $-arr

lar dw 1 dup(?)

sma dw 1 dup(?)

data ends

code segment

start:mov ax,data

mov ds,ax

lea si,arr

mov cx,len

shr cx,1

mov ax,0000h

mov bx,arr[si]

l1:cmp ax,arr[si]

jnc l2

mov ax,arr[si]

l2:cmp bx,arr[si]

jnc l3

mov bx,arr[si]

l3:inc si

inc si

loop l1

mov lar,ax

mov sma,bx

mov ah,4ch

int 21h

code ends

end start



-----------------------------------------------

assume cs:code,ds:data

data segment

arr db 99h,12h,56h

len dw $-arr

ev dw 1 dup(?)

data ends

code segment

start:mov ax,data

mov ds,ax

lea si,arr

mov cx,len

mov bx,0000h

l1:mov al,arr[si]

and al,01h

cmp al,00h

jnz l2

mov ah,00h

mov al,arr[si]

add bx,ax

l2:inc si

loop l1

mov ev,bx

mov ah,4ch

int 21h

code ends

end start



-----------------------------------------------------

assume cs:code,ds:data

data segment

arr db 99h,7ah,0b1h,33h,23h,52h

len dw $-arr

data ends

code segment

start:mov ax,data

mov ds,ax

lea si,arr

mov ax,len

l1:mov dI,si

inc di

mov cx,ax

dec cx

l2:mov bh,arr[si]

mov bl,arr[di]

cmp bh,bl

jc l3

mov arr[si],bl

mov arr[di],bh

l3:inc di

loop l2

inc si

dec ax

cmp ax,0001h

jnz l1

mov ah,4ch

int 21h

code ends

end start



---------------------------------------------------------

ASSUME CS: CODE, DS: DATA

DATA SEGMENT

STR DB 100 dup('$')

LEN DW 0000H

CAPITAL DB 1 DUP(?)

SMALL DB 1 DUP(?)

MSG1 DB 0DH, 0AH, "Capital letters: ","$"

MSG2 DB 0DH, 0AH, "Small letters: ","$"

DATA ENDS

CODE SEGMENT

START:MOV AX, DATA

MOV DS, AX

LEA SI, STR

INPUT: MOV AH,1H

INT 21H

CMP AL,13

JE L0

MOV [SI], AL

INC SI

INC LEN

JMP INPUT

L0:

MOV DL, 00

MOV DH, 00

LEA SI, STR

MOV CX, LEN

L1: MOV AL, STR [SI]

CMP AL, "a"

JNC L2

CMP AL, "A"

JNC L3

JC L6

L2: CMP AL, "z"

JC L4

JNC L6

L3: CMP AL, "Z"

JC L5

JNC L6

L4: INC DL

JMP L6

L5: INC DH

L6: INC SI

LOOP L1

MOV CAPITAL, DH

MOV SMALL, DL

MOV DX, OFFSET MSG1

MOV AH, 09H

INT 21H

MOV DL, CAPITAL

MOV AH, 02H

MOV BL, 30H

ADD BL, DL

MOV DL, BL

INT 21H

MOV DX, OFFSET MSG2

MOV AH, 09H

INT 21H

MOV DL, SMALL

MOV AH, 02H

MOV BL, 30H

ADD BL, DL

MOV DL, BL

INT 21H

MOV AH, 4CH

INT 21H

CODE ENDS

END START



---------------------------------------------------------

ASSUME CS: CODE, DS: DATA

DATA SEGMENT

STR DB 100 dup('$')

LEN DW $-STR

MSG1 DB 0DH, 0AH, "Pallindrome","$"

MSG2 DB 0DH, 0AH, "Not Pallindrome","$"

DATA ENDS

CODE SEGMENT

START:MOV AX, DATA

MOV DS, AX

LEA SI, STR

INPUT: MOV AH,1H

INT 21H

CMP AL,13

JE L1

MOV [SI], AL

INC SI

JMP INPUT

L1: MOV DI,OFFSET STR

DEC SI

L2: CMP SI,DI

JL L4

MOV AL,STR[SI]

MOV BL,STR[DI]

CMP AL,BL

JNE L3

DEC SI

INC DI

JMP L2

L3: MOV DX,OFFSET MSG2

MOV AH,09H

INT 21H

JMP L5

L4: MOV DX,OFFSET MSG1

MOV AH,09H

INT 21H

L5: MOV AH, 4CH

INT 21H

CODE ENDS

END START

