# 8 BIT ARITHMETIC

## ADDITION

DATA SEGMENT

A DB 09H

B DB 02H

C DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV AL,A

MOV BL,B

ADD AL,BL

MOV C,AX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 8 BIT ARITHMETIC

## SUBTRACTION

DATA SEGMENT

A DB 09H

B DB 02H

C DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV AL,A

MOV BL,B

SUB AL,BL

MOV C,AX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 8 BIT ARITHMETIC

## MULTIPLICATION

DATA SEGMENT

A DB 09H

B DB 02H

C DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV AX,0000H

MOV BX,0000H

MOV AL,A

MOV BL,B

MUL B

MOV C,AX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 8 BIT ARITHMETIC

## DIVISION

DATA SEGMENT

A DB 28H

B DB 02H

C DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV AX,0000H

MOV BX,0000H

MOV AL,A

MOV BL,B

DIV B

MOV C,AX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 16 BIT ARITHMETIC

## ADDITION

DATA SEGMENT

A DW 0202H

B DW 0408H

C DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV AX,A

MOV BX,B

ADD AX,BX

MOV C,AX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 16 BIT ARITHMETIC

## SUBTRACTION

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

N1 DW 0FFFFH

N2 DW 4567H

N3 DW ?

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV AX,N1

MOV BX,N2

SUB AX,BX

MOV N3,AX

LEA SI,N3

MOV AH,4C

INT 21H

CODE ENDS

END START

# 16 BIT ARITHMETIC

## MULTIPLICATION

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

N1 DW 4444H

N2 DW 4567H

N3 DD ?

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV AX,N1

MOV BX,N2

MUL BX

LEA SI,N3

MOV [SI],AX

MOV [SI+2],DX

MOV AH,4C

INT 21H

CODE ENDS

END START

# 16 BIT ARITHMETIC

## DIVISION (WORD BY BYTE)

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

N1 DW 0444H

N2 DB 45H

N3 DW ?

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV AX,N1

MOV BL,N2

DIV BL

MOV N3,AX

LEA SI,N3

MOV AH,4C

INT 21H

CODE ENDS

END START

# **32 BIT ARITHMETIC**

## **ADDITION**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

DATA1 DW 1234H

DATA2 DW 5555H

DATA3 DW 6789H

DATA4 DW 1111H

ANS1 DW 0000H

ANS2 DW 0000H

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV BX,DATA2

ADD BX,DATA4

MOV ANS1,BX

MOV CX,DATA1

ADC CX,DATA3

MOV ANS2,CX

MOV AH,4CH

INT 21H

CODE ENDS

END START

# **32 BIT ARITHMETIC**

## **SUBTRACTION**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

DATA1 DW 1234H

DATA2 DW 5555H

DATA3 DW 6789H

DATA4 DW 1111H

ANS1 DW 0000H

ANS2 DW 0000H

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV BX,DATA2

SUB BX,DATA4

MOV ANS1,BX

MOV CX,DATA1

SBB CX,DATA3

MOV ANS2,CX

MOV AH,4CH

INT 21H

CODE ENDS

END START

# BIT MANUPILATION

## DATA IS POSITIVE OR NEGATIVE

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

MSG1 DB ‘ENTERED NUMBER IS POSITIVE. $’

MSG2 DB ‘ENTERED NUMBER IS NEGATIVE. $’

INPUT DB ?

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV AL, INPUT

ROL AL, 01H

JC NEXT

LEA DX, MSG1

MOV AH, 09H

INT 21H

JMP LAST

NEXT: LEA DX, MSG2

MOV AH, 09H

INT 21H

LAST: MOV AH, 4CH

INT 21H

CODE ENDS

END START

# BIT MANUPILATION

## IF THE DATA IS ODD OR EVEN

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

MSG1 DB ‘ENTERED NUMBER IS ODD. $’

MSG2 DB ‘ENTERED NUMBER IS EVEN. $’

INPUT DB ?

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV AL, INPUT

SAR AL, 01H

JC NEXT

LEA DX, MSG2

MOV AH, 09H

INT 21H

JMP LAST

NEXT: LEA DX, MSG1

MOV AH, 09H

INT 21H

LAST: MOV AH, 4CH

INT 21H

CODE ENDS

END START

# BIT MANUPILATION

## COUNT THE NUMBER OF 1’S AND 0’S IN GIVEN DATA

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

MOV CX, 0008H

MOV AL, 24H

MOV BL, 00H

MOV DL, BL

NEXT: SAR AL, 01H

JC DOWN

INC BL

LOOP NEXT

JMP LAST

DOWN: INC DL

LOOP NEXT

LAST:MOV AH, 4CH

INT 21H

CODE ENDS

END START

**LARGEST/SMALLEST ELEMENT IN AN ARRAY**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

ARRAY DB 10H, 20H, 30H, 40H, 50H

COUNT DW 0005H

RESULT DB 00H

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS, AX

LEA SI, ARRAY

DEC COUNT

MOV CX, COUNT

MOV AL, [SI]

NEXT: INC SI

CMP AL, [SI]

JC DOWN

LOOP NEXT

JMP LAST

DOWN: XCHG AL, [SI]

LOOP NEXT

MOV RESULT, AL

LAST: MOV AH, 4CH

INT 21H

CODE ENDS

END START

**SORTING ARRAY IN ASCENDING/DESCENDING ORDER**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

Array DW 2233h, 8899h, 6677h, 0011h, 4455h

Count DW 0005h

DATA ENDS

CODE SEGMENT

START:

MOV AX, DATA

MOV DS, AX

MOV CX, Count

LEA SI, Array

NEXT: MOV BX, [SI]

INC SI

INC SI

CMP BX, [SI]

JNC DOWN

LOOP NEXT

DOWN: XCHG BX, [SI]

DEC SI

DEC SI

MOV [SI], BX

LOOP NEXT

MOV AH, 4Ch

INT 21h

CODE ENDS

END START

**CHARACTER SEARCH IN A STRING**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

String DB 'BMS COLLEGE'

Length EQU ($-String)

Key DB 'X'

Dis1 DB '-IS PRESENT IN GIVEN STRING$'

Dis2 DB '-IS NOT PRESENT IN GIVEN STRING$'

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS, AX

MOV ES, AX

MOV DL, Key

MOV AH, 02h

INT 21h

LEA DI, String

MOV AL, Key

MOV CX, Length

REPNE SCASB

JE PRESENT

LEA DX, Dis2

CALL Display

JMP OVER

PRESENT: LEA DX, Dis1

CALL Display

OVER: MOV AH, 4Ch

INT 21h

Display PROC NEAR

MOV AH, 09h

INT 21h

RET

Display ENDP

CODE ENDS

END START

**REVERSE OF A STRING:**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

MSG1 DB 0AH,0DH,"ENTER THE STRING: $"

MSG2 DB 0AH,0DH,"THE REVERSE :$"

STR1 DB 20 DUP("?")

REV DB 20 DUP("?")

DATA ENDS

DISPLAY MACRO MSG

MOV AH,09H

LEA DX,MSG

INT 21H

ENDM

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

DISPLAY MSG1

MOV CX,0000H

LEA SI,STR1

LOOP1: MOV AH,01H

INT 21H

MOV [SI],AL

CMP AL,0DH

JE LOOP2

INC SI

INC CX

JMP LOOP1

LOOP2: MOV BL,"$"

MOV [SI],BL

MOV DI,SI

LEA SI,REV

DEC DI

LOOP3: MOV BL,[DI]

MOV [SI],BL

DEC DI

INC SI

DEC CX

JNZ LOOP3

MOV BL,"$"

MOV [SI],BL

DISPLAY MSG2

DISPLAY REV

MOV AH,4CH

INT 21H

CODE ENDS

END START

**CHECK WHETHER A STRING IS PALINDROME OR NOT:**

DATA SEGMENT

MSG1 DB 0AH,0DH,"ENTER THE STRING: $"

MSG2 DB 0AH,0DH,"THE STRING IS PALINDROME:$"

MSG3 DB 0AH,0DH,"THE STRING IS NOT PALINDROME:$"

STR1 DB 50 DUP("?")

DATA ENDS

DISPLAY MACRO MSG

MOV AH,09H

LEA DX,MSG

INT 21H

ENDM

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

DISPLAY MSG1

MOV CX,0000H

LEA SI,STR1

LOOP1: MOV AH,01H

INT 21H

MOV [SI],AL

CMP AL,0DH

JE LOOP2

INC SI

INC CX

JMP LOOP1

LOOP2: DEC SI

LEA DI,STR1

LOOP3: MOV BL,[DI]

MOV AL,[SI]

CMP AL,BL

JNZ LOOP4

INC DI

DEC SI

DEC CX

JNZ LOOP3

DISPLAY MSG2

JMP LOOP5

LOOP4: DISPLAY MSG3

LOOP5: MOV AH,4CH

INT 21H

CODE ENDS

END START