

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
    INT 3
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
    INT 3
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

INT 3

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

INT 3

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

INT 3

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

INT 3

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

INT 3

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

INT 3

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