一、数据处理与代码转换

程序设计的基本任务是进行数据处理，数据处理的具体内容包括数据运算、存储、检索、排序、输入和输出等操作。在数据处理过程中，往往由于不同的应用目的，需要将数据按不同的编码格式进行编码和存储，在微型计算机中常用的数据编码有二进制编码、BCD 码和 ASCII 码。二进制编码是可以直接进行算术运算的编码，汇编语言中的四种基本运算指令 ADD、SUB、MUL 和 DIV 都可以单独地直接对二进制数进行运算，其结果还是二进制数。BCD 码是用 4 位二进制数表示一位十进制数的编码，其编码范围为 0000B～1001B，分别表示十进制数的 0～9。BCD 码是为了便于在某些场合中，将数据按十进制数进行运算、存储和输出所设定的一种数据格式编码。ASCII 码是用于数据的输入和输出的一种数据编码格式，即主机与计算机外部设备进行信息交换的一种数据编码格式。由于计算机在对某一具体的数据处理过程中往往包括数据的输入、存储、运算、输出等操作，在这些操作中，每一种操作所需要的数据格式往往不同于其它操作所需要的数据格式。例如输入的数据是 ASCII 码，而存储时可能要转换为二进制编码格式以便于运算，而运算的最后结果要输出，则又需要按 ASCII 码格式来表示。上述的过程表明，在数据处理过程中，往往要进行数据格式之间的转换，以适应于不同的数据处理操作。我们将两种数据格式之间的转换问题称之为代码转换。常用的代码转换有 BCD 码转二进制，二进制转 BCD 码；ASCII 码转 BCD 码，BCD 码转 ASCII 码；二进制转 ASCII 码，ASCII 码转二进制等。

本章选取了数据处理和代码转换方面最常用的一些实例，为便于用户直接在自己的源程序中引用这些实例，大多数的实例都以过程(子程序)的形式给出，少数实例以完整的源程序形式给出。

1　如何实现多位二进制加法

程序名	MBINADD. ASM	程序类别	汇编语言子程序
功　能	实现以字为单位的多位二进制数的加法		
使用说明	入口参数：DS：SI→加数地址，DS，DI→被加数地址 出口参数：DS，BX→和的存放地址		

(1) 程序说明

本程序具有将两个多位二进制数进行加法运算的功能。程序在进行加法运算时，每一次将加数与被加数对应的一个字进行加运算。多位数的长度由常数 COUNT(以字为单位)控制，所有三个数都含有 16×COUNT 个位，存放在 COUNT 个字长度的内存单元中。

(2) 程序清单

```
MBINADD     PROC    NEAR
            PUSH    SI
            PUSH    DI
            PUSH    BX
            PUSH    CX
            PUSH    AX
            MOV     CX,COUNT        ;加法运算的字计数值→CX
            CLC                     ;清进位标志
MBINADD1:
            MOV     AX,[SI]         ;第一个数
            INC     SI
            INC     SI
            ADC     AX,[DI]         ;加上第二个数
            INC     DI
            INC     DI
            MOV     [BX],AX         ;存储中间结果
            INC     BX
            INC     BX
            LOOP    MBINADD1        ;循环
            POP     AX
            POP     CX
            POP     BX
            POP     DI
            POP     SI
            RET
MBINADD     ENDP
```

2 如何实现多位二进制减法

程序名	MBINSUB. ASM	程序类别	汇编语言子程序
功　能	实现以字为单位的多位二进制数的减法		
使用说明	入口参数:DS:SI→被减数,DS:DI→减数,DS:BX→差的存放地址 出口参数:DS:BX→差的存放地址		

(1) 程序说明

本程序具有将两个多位二进制数进行减法运算的功能。程序在进行减法运算时，每一次将被减数与减数对应的一个字进行减运算。多位数的长度由常数 COUNT(以字为单位)控制，所有三个数都含有 16×COUNT 个位，存放在 COUNT 个字长度的内存单元中。

(2) 程序清单

```
MBINSUB   PROC    NEAR
          PUSH    SI
          PUSH    DI
          PUSH    BX
          PUSH    CX
          PUSH    AX
          MOV     CX,COUNT        ;减法运算的字长度计数→CX
          CLC                     ;清进位标志
MBINSUB1:
          MOV     AX,[SI]         ;取被减数的一个字
          INC     SI
          INC     SI
          SBB     AX,[DI]         ;减去减数的对应字
          INC     DI
          INC     DI
          MOV     [BX],AX         ;存差
          INC     BX
          INC     BX
          LOOP    MBINSUB1        ;循环
          POP     AX
          POP     CX
          POP     BX
          POP     DI
          POP     SI
          RET
MBINSUB   ENDP
```

3 如何实现多位二进制乘法

程序名	MBINMUL. ASM	程序类别	汇编语言子程序
功　能	实现以字为单位的多位二进制数的乘法		
使用说明	入口参数:DS:SI→乘数,DS:DI→被乘数,DS:BX→积的存放地址 出口参数:DS:BX→积的存放地址		

(1) 程序说明

本程序能够完成两个多位的二进制数的乘法运算。程序的设计思想完全模仿手工计算的过程,即取乘法的一个字(从低字开始取),乘以被乘数的各个字,保存部分积;再取乘数中较高的一个字,乘以被乘数的各个字,将本次的部分积向高地址方向移动一个字,加上前面运算结果的部分积;……,如此过程,直到将乘数中最高的一个字乘完被乘数的各个字,并将最后一次的部分积向高地址方向移动一个字,加上前面所有积的结果,形成多位二进制数的积。

多位数的长度由常数 COUNT(以字为单位)控制,输入的两个数各有 16×COUNT 位,输出数精度加倍。输入存放在 COUNT 个字长度的内存单元中,输出存放在 2×COUNT 个字长度的内存单元中。

(2) 程序清单

```
MBINMUL     PROC    NEAR
            PUSH    SI
            PUSH    DI
            PUSH    BX
            PUSH    CX
            PUSH    AX
            PUSH    BX              ;保存结果单元首地址
            MOV     AX,0
            MOV     CX,2 * COUNT    ;乘积的字长度→CX
            CLD
MBINMUL1:
            MOV     [BX],AX         ;初始化乘积单元
            INC     BX
            INC     BX
            LOOP    MBINMUL1
            POP     BX              ;恢复结果单元首地址
            MOV     CX,COUNT        ;被乘数的字长度→CX
MBINMUL2:
```

```
        PUSH    CX
        MOV     DX,[SI]         ;取乘数的一个字
        INC     SI
        INC     SI
        PUSH    BX
        PUSH    DI
        MOV     CX,COUNT
MBINMUL3:
        PUSH    CX
        PUSH    DX
        MOV     AX,[DI]         ;取被乘数的一个字
        INC     DI
        INC     DI
        MUL     DX              ;进行乘法运算
        ADD     [BX],AX         ;处理本次积的低字
        INC     BX
        INC     BX
        ADC     [BX],DX         ;处理本次积的高字
        POP     DX
        POP     CX
        LOOP    MBINMUL3        ;循环,用被乘数的每一字乘以乘数的当前字
        POP     DI
        POP     BX
        INC     BX              ;修改乘数指针
        INC     BX
        POP     CX
        LOOP    MBINMUL2        ;循环,准备用乘数的下一字乘以被乘数的每
                                ;一字
        POP     AX
        POP     CX
        POP     BX
        POP     DI
        POP     SI
        RET
MBINMUL ENDP
```

4 如何实现多位二进制除法

程序名	MBINDIV. ASM	程序类别	完整的汇编语言程序
功　能	实现以字为单位的多位二进制数的除法		
使用说明	入口参数:DIVIDED→被除数,DIVISOR→除数,REMAIND→商 出口参数:DIVIDED→余数,REMAIND→商		

(1) 程序说明

本程序能够完成两个多位二进制数的除法运算。程序设计的基本思想是,用被除数减去除数的方法实现除法运算。运算开始时,首先比较被除数与除数的大小,如果被除数小于除数,则返回 DOS,错误号为 01。如果被除数大于除数,则从被除数中减去除数,并将商加 1,然后循环,直到被除数小于除数时结束运算。这样,原来存放被除数的单元中得到的是余数,而商则存放在 REMAIND 开始的内存单元中,所有的数都按低字在低地址,高字在高地址方式存放。为便于用户直接应用,本例以源程序的形式给出。

多位数的长度由常数 COUNT(以字为单位)控制。被除数存放在 DIVIDED 开始的内存单元中,除数存放在 DIVISOR 开始的内存单元中,商存放在 REMAIND 开始的内存单元中。被除数、除数、余数及商的长度都是 COUNT 个字。

(2) 程序清单

```
CODE        SEGMENT
            ASSUME   CS:CODE,DS:CODE,ES:CODE
COUNT       EQU      2                ;被除数、除数的长度(以字为单位)
DIVIDED     DW       60000,1          ;被除数
DIVISOR     DW       50000,0          ;除数
REMAIND     DW       0,0              ;商
START:      PUSH     CS
            PUSH     CS
            POP      DS
            POP      ES
            MOV      SI,OFFSET DIVIDED
            MOV      DI,OFFSET DIVISOR
            MOV      BX,OFFSET REMAIND
            CALL     DIVCMP           ;比较被除数和除数
            JAE      NEXT             ;被除数大于等于除数时,转
            MOV      AX,4C01H         ;错误返回
            INT      21H
```

```
NEXT:   CALL    DIVSUB          ;从被除数中减去除数
        PUSH    BX
        MOV     CX,COUNT
        STC                     ;置进位标志,表示当前商要加 1
CACULA: ADC     WORD PTR [BX],0
        INC     BX
        INC     BX
        LOOP    CACULA          ;循环处理商的进位
        POP     BX
        CALL    DIVCMP          ;比较被除数与除数的大小
        JAE     NEXT            ;被除数大于等于除数时转
        MOV     AX,4C00H        ;正确返回 DOS
        INT     21H
        ;比较字符串子程序
DIVCMP  PROC    NEAR
        PUSH    SI
        PUSH    DI
        PUSH    CX
        ADD     SI,2*COUNT-2    ;SI 指向被除数的最后一个字
        ADD     DI,2*COUNT-2    ;DI 指向被除的最后一个字
        MOV     CX,COUNT        ;比较的字长度
        STD                     ;反向比较(从高地址到低地址方向)
        REPZ    CMPSW
        POP     CX
        POP     DI
        POP     SI
        RET
DIVCMP  ENDP
        ;字符串相减子程序
DIVSUB  PROC    NEAR
        PUSH    SI
        PUSH    DI
        PUSH    CX
        CLC
        MOV     CX,COUNT        ;相减的字计数值
DIVSUB1: MOV    AX,[DI]         ;取除数的一个字→AX
        INC     DI
        INC     DI
        SBB     [SI],AX         ;减去被除数中的对应字(考虑借位)
        INC     SI
```

```
            INC     SI
            LOOP    DIVSUB1
            POP     CX
            POP     DI
            POP     SI
            RET
DIVSUB      ENDP
CODE        ENDS
            END     START
```

5 如何将 BCD 码转换为二进制数

程 序 名	BCDTOBIN. ASM	程序类别	汇编语言子程序
功　　能	将 AX 寄存器中的 4 位 BCD 码转换成二进制数		
使用说明	入口参数：AX ←BCD 码，出口参数：AX←二进制数 注：引用该子程序时，数据段必须有存放权值的单元：W10		

(1) 程序说明

本程序具有将 AX 中的四位 BCD 码转换成二进制数的功能，转换的结果放在 AX 中。

由于一位 BCD 码用 4 位二进制数表示，这样一个字长的寄存器中最多能存放 4 位的 BCD 码。例如，AX＝9827H，如果被看作为 BCD 码，则其代表的数为九千八百二十七。

将 AX 中的 4 位 BCD 码转换为二进制数的过程可以用下述公式来描述：

((千位数×10＋百位数)×10＋十位数)×10＋个位数

引用该子程序时，数据段中必须有下列存放权值的单元：

```
W10     DW      10          ;十进制数权值
```

(2) 程序清单

```
AXBCDTO2    PROC    NEAR
                    PUSH    BX
                    PUSH    CX
                    PUSH    DX
                    MOV     BX,AX       ;保存 AX 中的 BCD 码到 BX
                    MOV     AX,0        ;结果单元清零
                    MOV     CX,4        ;共处理四位 BCD 码
RETRY:              PUSH    CX
                    MOV     CL,4
                    ROL     BX,CL       ;一位 BCD 码移到 BX 中的低半字节
```

```
        POP     CX
        MUL     W10             ;累加和乘以权值送 AX
        PUSH    BX
        AND     BX,0FH          ;屏蔽 BX 的高半字节
        ADD     AX,BX           ;累加下一位 BCD 码
        POP     BX
        LOOP    RETRY
        POP     DX
        POP     CX
        POP     BX
        RET
AXBCDTO2 ENDP
```

6 如何将二进制数转换为 BCD 码

程序名	BINTOBCD. ASM	程序类别	汇编语言子程序
功　能	将 AX 中的二进制数转换为 BCD 码		
使用说明	入口参数:AX→二进制数,出口参数:AX→BCD 码 注:引用该子程序时,数据段必须有存放权值的单元,W1000		

(1) 程序说明

本程序具有将 AX 中的二进制数转换成四位 BCD 码的功能,转换的结果放在 AX 中(设 AX 中的数值小于十进制数 10000)。

程序设计的思想是,首先将 AX 中的二进制数除以 1000 得到的商即是千位上的 BCD 码,将所得余数再除以 100 得到的商即百位上的 BCD 码,然后再将所得的余数除以 10 得到的商即是十位上的 BCD 码,最后所得的余数是个位上的 BCD 码。

引用该子程序时,数据段中必须有下列存放权值的单元:

```
W1000   DW      1000,100,10,1   ;十进制数千,百,十,个位上的权值
```

(2) 程序清单

```
AX2TOBCD  PROC    NEAR
          PUSH    SI
          PUSH    BX
          PUSH    CX
          PUSH    DX
          XOR     BX,BX                   ;BCD 码暂存单元清零
          MOV     SI,OFFSET W1000         ;权值首地址送 SI
```

```
                MOV     CX,4                ;循环次数 4 送 CX
RETRY:          PUSH    CX
                MOV     CL,4
                SHL     BX,CL
                MOV     DX,0
                DIV     WORD PTR [SI]       ;除以权值
                OR      BX,AX               ;组合 BCD 码
                MOV     AX,DX               ;余数送 AX
                POP     CX
                INC     SI
                INC     SI                  ;地址加 2,指向下一权值
                LOOP    RETRY
                MOV     AX,BX               ;BCD 码由 BX 送 AX
                POP     DX
                POP     CX
                POP     BX
                POP     SI
                RET
AX2TOBCD        ENDP
```

7 如何将 ASCII 码转换为 BCD 码

程 序 名	ASCTOBCD. ASM	程序类别	汇编语言子程序
功　能	将 ASCII 码转换为 BCD 码		
使用说明	入口参数:DS:SI→ASCII 码首地址,DS:BX→BCD 码首地址 出口参数:DS:BX→BCD 码首地址		

(1) 程序说明

本程序具有将 SI 所指的内存单元中存放的四个 ASCII 码转换成四位 BCD 码的功能,转换的结果放在 BX 所指的内存单元中。

由于两个 ASCII 码能组合成一字节的 BCD 码(两个 BCD 码),这样,四个 ASCII 码转换成 BCD 码,共能组合两个字节的 BCD 码。程序设计时,采用了两重循环,内循环执行一次可以把两个 ASCII 码的高 4 位屏蔽掉,将它们的低 4 位组合成一字节的 BCD 码。这样,外循环执行两次即可把四个 ASCII 码组合成两字节的 BCD 码。

需要指出的是,在数据段中定义的 BCD 码缓冲区的长度是 ASCII 码缓冲区长度的二分之一。

(2) 程序清单

```
ASCTOBCD PROC   NEAR
                PUSH    BX
                MOV     CX,2                ;装配两字节的 BCD 码
ATOB1:          PUSH    CX
                MOV     CX,2                ;内循环计数,一字节装配两个
                                            BCD 码
ATOB2:          PUSH    CX
                MOV     CL,4
                SHL     BYTE PTR [BX],CL    ;低 4 位清零,以便组合下一
                                            BCD 码
                MOV     AL,[SI]             ;取一 ASCII 码送 AL
                SUB     AL,30H              ;转换为 BCD 码
                OR      [BX],AL             ;组合一个 BCD 码
                INC     SI                  ;ASCII 码地址加 1
                POP     CX
                LOOP    ATOB2               ;一字节 BCD 码未组合完,转
                                            ATOB2
                INC     BX
                POP     CX
                LOOP    ATOB1               ;两字节的 BCD 码未组合完,转
                                            ATOB1
                POP     BX
                RET
ASCTOBCD ENDP
```

8 如何将 BCD 码转换为 ASCII 码

程 序 名	BCDTOASC. ASM	程序类别	汇编语言子程序
功　　能	将 BCD 码转换为 ASCII 码		
使用说明	入口参数:AL→BCD 码,BX→ASCII 码首地址 出口参数:BX→ASCII 码首地址		

(1) 程序说明

本程序具有将 AL 中的两位 BCD 码换成相应的 ASCII 码的功能,转换的结果存放在 BX 所指向的连续的两字节的内存单元中。

程序设计的思想是，将 AL 中的两个 BCD 码进行分离，转换为两个分离的 BCD 码，然后分别加上 30H 形成 ASCII 码。

注意：在数据段定义的 ASCII 码缓冲区的长度是 BCD 码缓冲区长度的两倍。

(2) 程序清单

```
BCDTOASC PROC    NEAR
                 PUSH    CX
                 PUSH    BX
                 MOV     CX,2            ;一字节需处理 2 次
BTOA1:           PUSH    CX
                 MOV     CL,4
                 ROL     AL,CL           ;左环移 4 位
                 PUSH    AX
                 AND     AL,0FH          ;截取低 4 位
                 OR      AL,30H          ;0~9 变 ASCII 码
                 MOV     [BX],AL
                 INC     BX              ;ASCII 码缓冲区地址加 1
                 POP     AX
                 POP     CX
                 LOOP    BTOA1
                 POP     BX
                 POP     CX
                 RET
BCDTOASC  ENDP
```

9 如何将二进制数转换为 ASCII 码

程序名	BINTOASC. ASM	程序类别	汇编语言子程序
功　能	将 AX 中的二进制数转换为 ASCII 码		
使用说明	入口参数：AX←二进制数 出口参数：SI←ASCII 码首地址		

(1) 程序说明

本程序具有将 AX 中的二进制数转换成 ASCII 码的功能，转换的结果存放在从 ASCBUF 开始的连续的 5 个内存单元中。

AX 中的二进制数最大数值为 65535，转换为 ASCII 码需 5 个字节单元。程序的设计思想是，首先将 AX 中的数除以 10，所得余数为个位上的数，加上 30H 变为相应的 ASCII 码；所得

的商再作为被除数除以 10，得到的余数为十位上的数，加上 30H 变为相应的 ASCII 码；所得的商再作为被除数除以 10，得到的余数为百位上的数，……，直到被除数小于 10 时，得到最后的一位数。

引用该子程序时，数据段中必须具有下列存放 ASCII 码的缓冲区：

```
ASCBUF   DB   5 DUP(0)
```

(2) 程序清单

```
BINTOASC PROC     NEAR
                  MOV     CX,10          ;除数送 CX
                  LEA     SI,ASCBUF+4    ;SI 指向 ASCII 码个位数地址
BTOA1:            CMP     AX,10          ;二进制数小于 10 吗?
                  JB      BTOA2          ;小于 10 转 BTOA2
                  XOR     DX,DX          ;被除数高字清零
                  DIV     CX             ;除 10
                  OR      DL,30H         ;余数变 ASCII 码
                  MOV     [SI],DL        ;存一字节 ASCII 码
                  DEC     SI             ;ASCII 码地址减 1
                  JMP     BTOA1
BTOA2:            OR      AL,30H
                  MOV     [SI],AL        ;存最高位的 ASCII 码
                  RET
BINTOASC  ENDP
```

10　如何将 ASCII 码转换为二进制数

程 序 名	ASCTOBIN. ASM	程序类别	汇编语言子程序
功　　能	将 ASCII 码转换为二进制数		
使用说明	入口参数：ASCBUF→ASCII 码，出口参数：BINVAL→二进制数 使用的内存单元：ASCLEN 中存放 ASCII 码字符个数，MULT10 存放权值		

(1) 程序说明

本程序具有将 ASCBUF 开始的内存单元中的 4 个连续的 ASCII 码转换为二进制数的功能，转换的结果存放在 BINVAL 单元中。

程序设计的思想是，将 4 个 ASCII 的高 4 位分别屏蔽掉，各自转换为 0～9 的数，然后分别乘以各自的权值，最后累加起来即得到相应的二进制。

引用该子程序时，数据段要定义如下的内存单元：

```
ASCBUF    DB   '6472'          ;ASCII 码字符串
```

```
ASCLEN   DB  $-ASCBUF   ;ASCII 码字符个数
BINVAL   DW  0           ;二进制数结果
MULT10   DW  1           ;十进制权值存放单元
```

(2) 程序清单

```
ASCTOBIN  PROC    NEAR
          MOV     CX,10          ;乘法因子送 CX
          LEA     SI,ASCBUF-1    ;ASCII 码首地址减 1 送 SI
          MOV     BX,ASCLEN      ;ASCII 码字符个数送 BX
ATOB1:    MOV     AL,[SI+BX]     ;取一个 ASCII 码(从后往前)
          AND     AX,000FH       ;转二进制数
          MUL     MULT10         ;乘当前权值
          ADD     BINVAL,AX      ;送累加和单元
          MOV     AX,MULT10
          MUL     CX             ;计算下一位的权值
          MOV     MULT10,AX      ;改变当前权值
          DEC     BX             ;计数减 1
          JNZ     ATOB1          ;未处理完,转 ATOB1
          RET
ASCTOBIN  ENDP
```

11 如何将十六进制数转换为十进制数

程序名	HEXTODEC. ASM/C	程序类别	完整的汇编/C 语言程序
功　能	将键盘输入的 0～0FFFFH 之间的十六进制数转换为十进制数并显示出来		
使用说明	汇编、连接以后可直接运行,按屏幕提示信息操作使用		

(1) 程序说明

在进行汇编语言程序设计和软件开发过程中,经常会遇到要将一个 4 位的十六进制数转换为相应的十进制数的问题,如果被转换的数比较小,口算还是笔算都比较容易,但如果要对一个较大的数进行转换,则无论是口算和笔算都容易出错,又浪费时间,为了解决这一问题,我们编写一个专用的程序来完成这一转换任务。

本例中的程序允许用户从键盘上输入一个 0～FFFFH 之间的任意一个 4 位的十六进制数(其中 A～F 可以为大写,也可以是小写),然后程序将其转换为对应的十进制数显示出来。在程序中,主要调用了 HEXIBIN 和 BINIDEC 两个子程序。子程序 HEXIBIN 是将十六进制数转换为二进制数的子程序,而子程序 BINIDEC 则是将二进制数转换为十进制数的子程序。

为了转换一批数据，该程序被设计成循环运行的形式，在转换完一个十六进制数以后，继续处于接收下一个十六进制数的状态，直至遇到〈Esc〉键才返回 DOS。

(2) 程序清单

```
CODE        SEGMENT
            ASSUME    CS:CODE
HEXIDEC     PROC      FAR
            PUSH      DS
            SUB       AX,AX
            PUSH      AX
START:      CALL      HEXIBIN     ;接收输入的十六进制数并转换为二进制数
            CMP       AL,1BH      ;是 ESC 键,转出口
            JZ        EXIT
            CALL      CRLF        ;回车、换行
            CALL      BINIDEC     ;将二进制数转换为十进制数
            CALL      CRLF        ;回车、换行
            JMP       START
EXIT:       RET
HEXIDEC     ENDP
HEXIBIN     PROC      NEAR
            MOV       BX,0        ;累加和单元初始化
INHEX:      MOV       AH,1
            INT       21H         ;接收键盘输入
            CMP       AL,1BH
            JZ        HEXEND      ;是 ESC 键,转出口
            SUB       AL,30H
            JL        HEXEND      ;输入的字符小于"0",转出口
            CMP       AL,10
            JL        ADD_TO      ;是"0"~"9"之间的数字,转
            SUB       AL,7        ;将字母转换为对应的二进制数
            CMP       AL,10
            JL        HEXEND      ;是小于"A"的字母,转出口
            CMP       AL,16
            JL        ADD_TO      ;是"A"~"F"之间的字母,转
            SUB       AL,20H      ;处理小写字母
            CMP       AL,10
            JL        HEXEND      ;是小于"a"的字符,转出口
            CMP       AL,16
            JGE       HEXEND      ;是大于"f"的字符,转出口
ADD_TO:     MOV       CL,4
```

```
            SHL     BX,CL       ;将当前数字乘以权值
            MOV     AH,0
            ADD     BX,AX       ;累加
            JMP     INHEX
HEXEND:     RET
HEXIBIN     ENDP
BINIDEC     PROC    NEAR
            MOV     CX,10000
            CALL    DEC_DIV     ;显示万位上的数
            MOV     CX,1000
            CALL    DEC_DIV     ;显示千位上的数
            MOV     CX,100
            CALL    DEC_DIV     ;显示百位上的数
            MOV     CX,10
            CALL    DEC_DIV     ;显示十位上的数
            MOV     CX,1
            CALL    DEC_DIV     ;显示个位上的数
            RET
BINIDEC     ENDP
DEC_DIV     PROC    NEAR
            MOV     AX,BX
            MOV     DX,0
            DIV     CX
            MOV     BX,DX
            MOV     DL,AL
            ADD     DL,30H
            MOV     AH,2
            INT     21H
            RET
DEC_DIV     ENDP
CRLF        PROC    NEAR
            MOV     DL,0DH
            MOV     AH,2
            INT     21H
            MOV     DL,0AH
            MOV     AH,2
            INT     21H
            RET
CRLF        ENDP
CODE        ENDS
```

END HEXIDEC

下面是用 Turbo C 语言实现的十六进制数转十进制数的程序，该程序可以将双字长度的十六进制数转换为十进制数，如果输入的十六进制数在 7FFFFFFF～FFFFFFFF 之间时，转换的结果将是负数。

```
#include <stdio.h>
unsigned long int num16=0xf1527;
unsigned long int num10,num100;
long int main()
{long int temp0,temp1;
 int i,j;
Printf("\nNum16 is ",num16);
scanf("%lx",&num16);
temp0=(num16)28);
num10=temp0&0x0f;
for(i=1;i<=7;i++)
{ j=7-i;
  num10=num10*16;
  temp1=(num16)(4*j));
  temp1=temp1&0xf;
  num10=num10+temp1;
}
printf("\nNum10 is %ld.",num10);
return (num10);
}
```

12 如何将十进制数转换为十六进制数

程序名	DECTOHEX. ASM/C	程序类别	完整的汇编/C 语言程序
功　能	将键盘输入的 0～65535 之间的十进制数转换为十六进制数并显示出来		
使用说明	汇编、连接以后可直接运行，按屏幕提示信息操作使用		

(1) 程序说明

将十进制数转换为十六进制数的问题，也是实际工作中经常遇到的问题之一。下面的一个程序能够接收任何一个小于等于 65535 的十进制数，并显示出对应的十六进制数。程序在接收十进制数的过程中，只要输入任何一个非 0～9 之间的数程序即认为所输入的十进制数结