实验一　汇编程序的汇编及运行

1. 实验目的

（1）熟悉汇编程序的汇编、连接、执行过程；

（2）生成 LST 文件，查看 LST 文件；

（3）生成 OBJ 文件，修改语法错误；

（4）生成 EXE 文件；

（5）执行程序。

2. 实验涉及知识

汇编程序从编写到执行的过程

编程→. ASM→编译→. OBJ→连接→. EXE→加载→内存中的程序→执行

1）编写源程序

用 EDIT 或记事本输入各段，并存储成源程序（保存在 MASM 目录下），扩展名为 ASM。

2）对源程序进行汇编、连接

操作如下：

（1）在 DOS 状态下，进入 MASM 目录；

（2）输入命令：MASM 文件名（连同扩展名）；

（3）在系统出现的提示中输入：

object　filename[.obj]：按回车键。

Source　listling[nul.lst]：输入：文件主名，生成 lst 文件，也可以不生成，直接按回车键。Lst 文件用于查看编译为目标文件的过程中产生的中间结果。

Cross-reference [nul.crf]：按回车键。

（4）如果系统没有提示出错，那么编译过程就完成了。如果提示错误则应用 edit 打开源文件进行修改，然后再重复 2 和 3 步骤

（5）输入：link　文件主名（不要输扩展名，也可以输入扩展名.obj）

run file [.exe]：按回车键。

List file [nul.map]：按回车键。

Libraries [.lib]：按回车键。

（6）没有提示错误，汇编过程就结束了，产生了 exe 可执行文件。如果出现‘no　strack

segment’的错误警告，不用理会。

3）执行程序（由 DOS 中的 COMMAND 命令将文件加载入内存）

在 DOS 提示符下直接输入文件主名就可以执行文件了。程序执行后，有的会显示结果，有的可能执行后什么结果都没有，是因为程序中没有显示命令。

3. 实验内容

1）将下面的数据段输入，取名 1.ASM，保存在 MASM 文件夹下。生成 LST 文件，（不必连接、运行）用 EDIT 查看 1.LST 文件。试回答：DA1，DA2 的偏移量分别是多少？COUNT 的值为多少？

```
DATA    SEGMENT
ORG 20H
NUM1=9
NUM2=NUM1+10H
DA1   DB 'Tinkpad PC'
      DB 0AH, 0DH
COUNT EQU $-DA1
DATA    ENDS
END
```

```
                                                        Page     1-1


0000                            DATA    SEGMENT
0020                                    ORG 20H
= 0009                                  NUM1=9
= 0019                                  NUM2=NUM1+10H
0020  54 69 6E 6B 70 61 64                      DA1   DB 'Tinkpad PC'
      20 50 43
002A  0A 0D                                       DB 0AH, 0DH
= 000C                                  COUNT EQU $-DA1
002C                                    DATA    ENDS
                                        END
Microsoft (R) Macro Assembler Version 5.00                  1/12/10 12:48:54
                                                            Symbols-1
```

```
Segments and Groups:

                N a m e                 Length   Align   Combine Class

DATA . . . . . . . . . . . . . .        002C     PARA    NONE

Symbols:

                N a m e                 Type     Value   Attr

COUNT  . . . . . . . . . . . . .        NUMBER   000C

DA1  . . . . . . . . . . . . . .        L BYTE   0020    DATA

NUM1 . . . . . . . . . . . . . .        NUMBER   0009
NUM2 . . . . . . . . . . . . . .        NUMBER   0019

@FILENAME  . . . . . . . . . . .        TEXT  1
```

Count 000C

DA1 0020

2）输入下面错误的文件，修改错误语句。（MASM 没有出现错误即可。不必连接、运行。）

```
Object filename [2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:
2.txt(4): Out of memory
```

将 VAR3　　DB ’ABCDEF’改为 VAR3　DB 'ABCDEF'

```
Object filename [2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:
2.txt(2): error A2009: Symbol not defined: ODH
2.txt(3): error A2009: Symbol not defined: A4H
2.txt(9): error A2009: Symbol not defined: DE
2.txt(10): error A2105: Expected: instruction or directive
2.txt(14): error A2035: Operand must have size
2.txt(15): error A2068: Cannot address with segment register
2.txt(16): error A2052: Improper operand type
2.txt(17): error A2068: Cannot address with segment register
2.txt(21): error A2009: Symbol not defined: START
```

正确代码：

```
DATA    SEGMENT
        VAR1    DB 0, 25, 0DH, 255
        VAR2    DB 12H, 04H, 6BH
        VAR3    DB 'ABCDEF'
        VAR4    DW 1234H, 5678H
        VAR5    DW 10H DUP(?)
        DATA    ENDS
        CODE    SEGMENT
                ASSUME      CS: CODE, DS: DATA
        START:  MOV     AX, DATA
                MOV     DS, AX
                LEA      SI, VAR5
                MOV     BX, OFFSET VAR2
                MOV     AX,0ABH
                         MOV     [SI],AX
                MOV     AL, VAR1+2
```

```
        MOV     AX,[SI]
                MOV     [BX],AX
        MOV     AX,VAR4
                MOV     VAR5+4,AX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

3）输入下面程序并运行

```
STACKS      SEGMENT    STACK
DW          128 DUP(?)
STACKS      ENDS
DATAS       SEGMENT
STRING      DB 'WELCOME!', 13, 10, '$'
DATAS       ENDS
CODES       SEGMENT
            ASSUME     CS: CODES, DS: DATAS
START:      MOV    AX, DATAS
            MOV    DS, AX
            LEA    DX, STRING
            MOV    AH, 9
            INT    21H
            MOV    AH, 4CH
            INT    21H
CODES       ENDS
            END    START
```

修正代码 STRING DB 'WELCOME!', 13, 10, '$' START: MOV AX, DATAS

```
D:\m?>masm 3.txt
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [3.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  50628 + 449164 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

```
D:\m?>link 3

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
```

实验二　顺序程序设计

1. 实验目的

（1）学习使用 DEBUG 的命令；

（2）使用 DEBUG 命令在数据段中查看程序运行的结果；

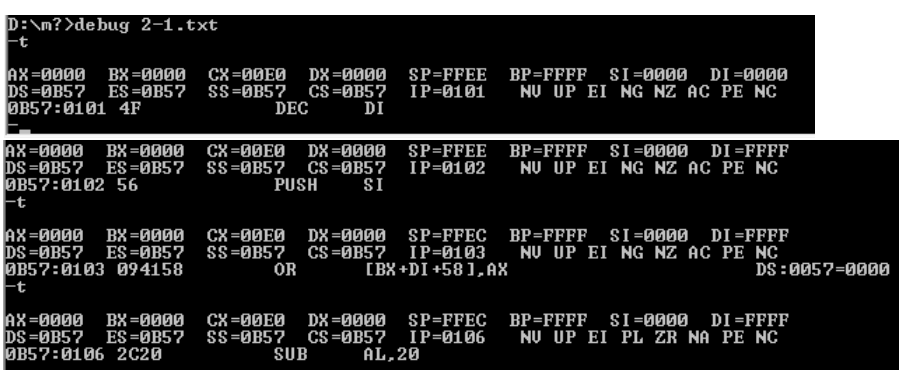
（3）利用 DEBUG 运行简单的程序段。

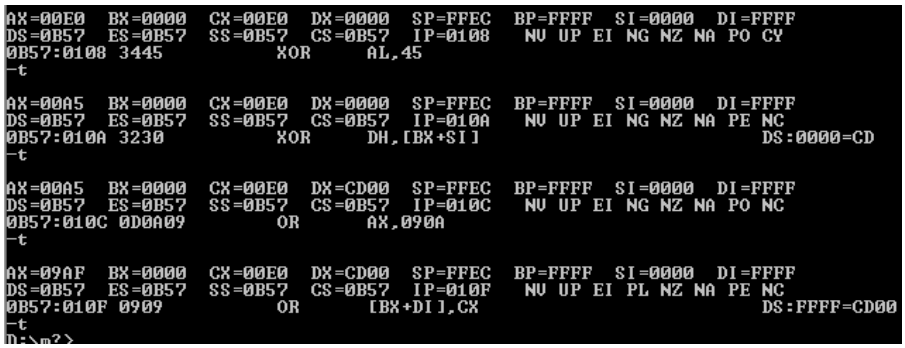
2. 实验内容

1）输入程序观察寄存器变化

使用 DEBUG，将下面的程序段写入内存，逐条执行，观察每条指令执行后，CPU 中相关寄存器的内容变化。注意用 T 命令执行时，CS: IP 寄存器的内容

```
MOV     AX, 4E20
ADD     AX, 1416
MOV     BX, 2000
ADD     AX, BX
MOV     BX, AX
ADD     AX, BX
MOV     AX, 001A
MOV     BX, 0026
ADD     AL, BL
ADD     AH, BL
ADD     BH, AL
MOV     AH, 0
ADD     AL, BL
ADD     AL, 9C
```





IP 值在不断增加

2）下列程序单步运行，注意 AL，BX，CX 寄存器的变化，并观察数据段字母的变化。如果是将小写字母改成大写字母带注释的语句该如何修改？

```
DSEG    SEGMENT
MSG1    DB  'abc'
DSEG    ENDS
CSEG    SEGMENT
ASSUME      CS: CSEG, DS: DSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        LEA     BX, MSG1
        MOV     CX, 3
S:      MOV     AL, [BX]
        AND     AL, 11011111B    ；将 AL 中的 ASCII 码的第 5 位置 0，
                                 ；变成大写字母。
        MOV     [BX], AL
        INC     BX
        LOOP    S
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
CSEG    ENDS
        END     START
```

3）程序的跟踪执行操作

在 DOS 下直接输入文件主名就可以执行文件了，有的程序会显示结果，可能执行后什么结果都没有，是因为程序中没有显示命令。那么如何查看程序的运行结果呢？

程序执行过程的跟踪操作步骤如下：

（1）在 DOS 下输入：DEBUG 文件名.EXE

（2）在 DEBUG 提示符下输入 U 命令

如果程序中有数据段，可以看到反汇编后第一句可执行语句为：

A 地址：B 地址　　MOV　AX, K 地址　　如：1261：0000　MOV　　AX, 1260

其中：K 地址就是数据段的段寄存器内容，A 地址为代码段段寄存器地址，B 地址为程序第一条指令的偏移地址。

（3）可以用 T 命令单步执行指令，执行到 MOV　AH, 4CH 时结束，也可以用 G 命令执行整个程序，输入：G=B 地址（如：G=0000）

（4）用 D 命令查看程序执行后数据段的变化

输入：D　K 地址：0　　（如：D1260：0）

在显示的数据中，对照源程序或 LST 文件查看结果所在的偏移地址的内容。

4）输入下面的程序，按实验一和上面的步骤运行一遍。这是一个两个数相与的程序。结果存放在 MSG2 单元中，偏移地址为？值为多少？

```
DSEG    SEGMENT
MSG1    DW 7856H, 2038H
MSG2    DW ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME      CS: CSEG, DS: DSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        MOV     AX, MSG1
        AND     AX, MSG1+2
        MOV     MSG2, AX
        MOV     AL, 0
        MOV     AH, 4CH
```

```
        INT     21H
CSEG    ENDS
        END     START
```

3.编写调试下面的程序，用 DEBUG 查看数据段中的结果

1）$Z=((W-X)*10+5)/(X+Y)$，X，Y，W 为字节类型变量，结果存于 Z 单元，写出数据段和代码段。

2）X，Y 为字节类型数，求 $Z=((X+Y)*8-(X-Y)*2)/16$，写出完整的数据段和代码段，不用乘除指令。

实验三　分支循环程序设计

1. 实验目的

（1）学习调试程序，查找逻辑错误；

（2）学习分支语句的编程和调试；

（3）学习循环语句的编程和调试。

2. 实验内容

1）有 10 个数，统计正数的个数，存放在变量 M 中中。经过汇编后，形成 EXE 文件。在 DEBUG 中，先用 G＝0 命令执行程序，用 D 命令查看 M 单元的内容，会发现结果不正确。用单步执行命令 T＝0，单步执行程序，查找程序中的逻辑错误，注意每一次循环 中 AL 寄存器中值的变化是否正确?（AL 寄存器中存放正数的个数）

```
DSEG    SEGMENT
MSG     DB   4, -2, -6, 0, 5, 67, 8, -3, 5, 6
M       DB   ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME      CS: CSEG, DS: DSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        MOV     CX, 10
        MOV     AL, 0
        LEA     SI, MSG
L1:     MOV     BL, [SI]
        CMP     BL, 0
        JBE     NEXT
        INC     AL
NEXT:   INC     SI
        LOOP    L1
        MOV     M, AL
```

```
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
CSEG    ENDS
        END     START
```

2）数据段中是一组无符号数，将最小数存放在 M 单元中。按上题方法查找一处逻辑错误。

```
DSEG    SEGMENT
MSG     DB    13, 15, 7, 25, 24
M         DB    ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME      CS: CSEG, DS: DSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        MOV     CX, 4
        MOV     AL, MSG
        MOV     SI, OFFSET MSG+1
L1:     CMP     AL, [SI]
        JB      NEXT
        MOV     AL, [SI]
NEXT:   LOOP    L1
        MOV     M, AL
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
        CSEG    ENDS
        END     START
```

3）编程：在首地址为 BUF 开始的内存单元中存有 10 个字节数，求其中 0 的个数，并将结果存于 RESULT 中。

4）编程：$Y = \sum A_i * B_i$，Ai，Bi 为字节型无符号数，分别存于 NUM1 和 NUM2 开始的连续存储单元中，结果存于 REST 单元中。

实验四　子程序设计

1. 实验目的

（1）学习子程序的编写，主子程序的调用；

（2）不同模块间程序的调用和调试。

2. 实验内容

1）数据段中的 3 个字符，调用子程序将其逐个显示出来。子程序的功能是显示一个字符。单步执行，对 CALL 语句和 RET 语句观察 SP，IP 的变化，并用 D 命令查看栈顶的内容。

```
DATA    SEGMENT
MAG     DB  'ABC'
DATA    ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     SI, 0
        MOV     CX, 3
LL:     MOV     DL, MAG[SI]
        CALL    MADD
        INC     SI
        LOOP    LL
        MOV     AH, 4CH
        INT     21H
MADD    PROC
        MOV     AH, 02H
        INT     21H
        RET
MADD    ENDP
CODE    ENDS
```

```
        END     START
```

2）阅读 S31.ASM 和 S32.ASM 两个模块中的程序，并分别汇编，然后连接成一个可执行文件 S31.EXE。

具体步骤如下：

MASM　S31.ASM（分别汇编）

MASM　S32.ASM

LINK　S31 S32　（将两个文件连接成为一个文件名为 S31）

S31.EXE　　（运行）

3）编程：利用主程序调用子程序，比较 BUF1 和 BUF2 缓冲区中不相等的字符，并将不相等的字符显示出来。（也可以将找到的不相等字符逐个显示，用 INT 21H 的 02 功能调用）

4）编程：子程序搜索指定字符缓冲区中是否有 N，如果有用 Y 替代。调用子程序将 BUF1，BUF2，BUF3 中的 N 全部用 Y 替代。

附源程序：

NAME S31.ASM

不同模块间的段间调用。从键盘输入 2 位非压缩 BCD 数，存入 AX 寄存器中。为了将其转换为二进制数，编写一个子程序 TRAN。显示子程序 DISP 完成将 16 进制数转换为对应的 ASCII 码并显示该字符，显示子程序在另一模块 S32 中。输入数据为 00 时结束。

```
        EXTRN   DISP: FAR
CODE    SEGMENT     PARA    'CODE'
        ASSUME      CS: CODE
START:  MOV     AH, 01H
        INT     21H
        MOV     BL, AL
        INT     21H
        MOV     AH, AL
        MOV     AL, BL
        CMP     AX, 3030H
        JE      EXIT
        CALL    NEAR PTR TRAN
```

```
        CALL    FAR   PTR disp
        JMP     START
EXIT:   MOV     AH, 4CH
        INT     21H
TRAN    PROC    NEAR          ；将输入的 ASCII 码转换成 2 进制数
        AND     AX, 0F0FH     ；将 ASCII 码转换成非压缩 BCD 码，
                              ；高位在 AL 中
        MOV     BL, AH
        MOV     CL, 10D
        MUL     CL
        ADD     AL, BL
        RET
TRAN    ENDP
CODE    ENDS
        END     START
NAME    S32.ASM
PUBLIC  DISP
CODE1   SEGMENT    PARA ‘CODE’
        ASSUME     CS: CODE1
DISP    PROC    FAR
        MOV     BL, AL
        MOV     BH, 00
        MOV     CH, 4
ROLL:   MOV     CL, 4
        ROL     BX, CL
        MOV     DL, BL
        AND     DL, 0FH
        CMP     DL, 9
        JBE     NEXT1
        ADD     DL, 07H
```

```
NEXT1:  ADD     DL, 30H
        MOV     AH, 02H
        INT     21H
        DEC     CH
        JNZ     ROLL
        RET
DISP    ENDP
CODE1   ENDS
        END
```

实验五　DOS 功能调用

1. 实验目的

（1）学会 DOS 中关于显示功能调用指令的用法；

（2）领会修改显存方法显示字符。

2. 实验内容

1）输入一个字符，显示出来。

2）输入一个字符串，显示出来。

3）BUF 开始的 3 个 16 位二进制数用十六进制数的形式显示出来。

（1）理解程序

（2）输入程序，汇编，运行（在 DOS 状态下输入：文件名.EXE），观察结果。

（3）如果要分行显示，程序要如何修改？

```
DATA    SEGMENT
BUF     DW 4F59H, 56A8H, 0FF90H
DATA    ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     SI, 0
P:      MOV     BX, BUF[SI]
        MOV     CH, 4
L:      MOV     CL, 4
        ROL     BX, CL
        MOV     DL, BL
        AND     DL, 0FH
        CMP     DL, 10
        JB      NEXT
        ADD     DL, 7
NEXT:   ADD     DL, 30H
```

```
        MOV     AH, 2
        INT     21H
        DEC     CH
        JNZ     L
        INC     SI
        INC     SI
        CMP     SI, 4
        JNA     P
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

4）直接改变显存内容显示。

（1）显存空间分配：

内存地址空间中，B8000H-BFFFFH 共 32KB 的空间，为 80×25 彩色字符模式显示缓冲区。向这个地址空间写入数据，写入的内容将立即出现在显示器上。

80×25＝2000 个字符，每个字符在缓冲区中占 2 个字节，一个字节存放 ASCII 码，一个字节存放字符属性（字符颜色、字符背景颜色、闪烁、高亮度）

（2）偏移地址计算：

如第 2 行，第 40 列：（2-1）×160＋40×2 这个偏移地址中存放字符的 ASCII 码，

（2-1）×160＋40×2＋1 这个偏移地址中存放字符的属性。

第 M 行，第 N 列的一般计算公式为：

（M-1）×160＋N*2 存放 ASCII 码

（M-1）×160＋N*2+1 存放属性

（3）运行下列程序，在屏幕的第 1 行，会显示一个字符 X。根据上面（1）、（2）点理解程序段。

```
STACK   SEGMENT
    DB  128  DUP (0)
STACK   ENDS
CSEG    SEGMENT
```

```
                ASSUME      CS: CSEG, SS: STACK
START:          MOV         AX, STACK
                MOV         SS, AX
                MOV         SP, 128
                MOV         AX, 0B800H
                MOV         ES, AX
                MOV         AH, ‘X’
S:              MOV         ES: [160*1+40*2], AH
                MOV         AL, 0
                MOV         AH, 4CH
                INT         21H
CSEG            ENDS
                END         START
```

5）编程显示以下图案。

****

****

*****

***

*

6）编写程序，统计字缓冲区中的 20 个数据的正数、负数、0 的个数，并将统计结果以 16 进制形式显示出来。

实验六　字符处理程序设计

1. 实验目的

（1）熟悉串操作指令的功能与应用；

（2）掌握串操作指令的寻址方式及使用方法，编写常用的字符串处理程序。

2. 实验内容

从键盘键入一个字符串，存入内存 BUF 为首地址的单元中，现要求统计其中含有小写字母的个数，并将统计结果以两位十进制数形式显示在屏幕上。

附参考源程序：

```
DATA    SEGMENT
BUF     DB 90 DUP(?)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, 0
        LEA     DI, BUF
AGAIN:  MOV     AH, 1
        INT     21H
        CMP     AL, 0DH
        JZ      DONE
        MOV     [DI], AL
        INC     DI
        INC     CX
        JMP     AGAIN
DONE:   MOV     DL, 0DH
        MOV     AH, 2
        INT     21H
        MOV     DL, 0AH
```

```
        INT     21H
        LEA     SI, BUF
        MOV     CH, 0
        MOV     BL, 0
        CLD
NEXT1:  LODSB
        CMP     AL, 61H
        JB      NEXT2
        CMP     AL, 7AH
        JA      NEXT1
        INC     BL
NEXT2:  LOOP    NTXT1
        MOV     AL, BL
        MOV     AH, 0
        MOV     CL, 10
        DIV     CL
        XCHG    AH, AL
        PUSH    AX
        MOV     DL,  AH
        OR      DL, 30H
        MOV     AH, 2
        INT     21H
        POP     AX
        MOV     DL, AL
        OR      DL, 30H
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    EDNS
        END     START
```

第 3 篇　课程设计辅导

实例 1 动画设计《甜蜜的生活》

1. 设计要求

1）了解并掌握汇编语言设计的一般方法，具备初步的独立分析和设计能力；

2）初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

3）提高综合运用所学的理论知识和方法独立分析和解决问题的能力；

2. 主要仪器设备（实验用的软硬件环境）

硬件环境：PC 机

软件环境：DOS 操作系统。

3. 设计内容

3.1 设计思路

整体：设计分为两个画面。

第一画面：男孩右手拿花向女孩跑去，这时天上忽然出现了一群小鸟，男孩向女孩求婚，女孩感动地接受了，迸出一颗心。

第二画面：两人过着幸福的生活。两人站在房子外，气球飘向了天。

3.2 设计总体结构图

总体结构图见图 3.1 所示。

3.3 设计流程图

1．静止程序流程图如图 3.2 所示。

2．移动程序的流程图如图 3.3 所示。

3.4 动画示意图如图 3.4 所示。

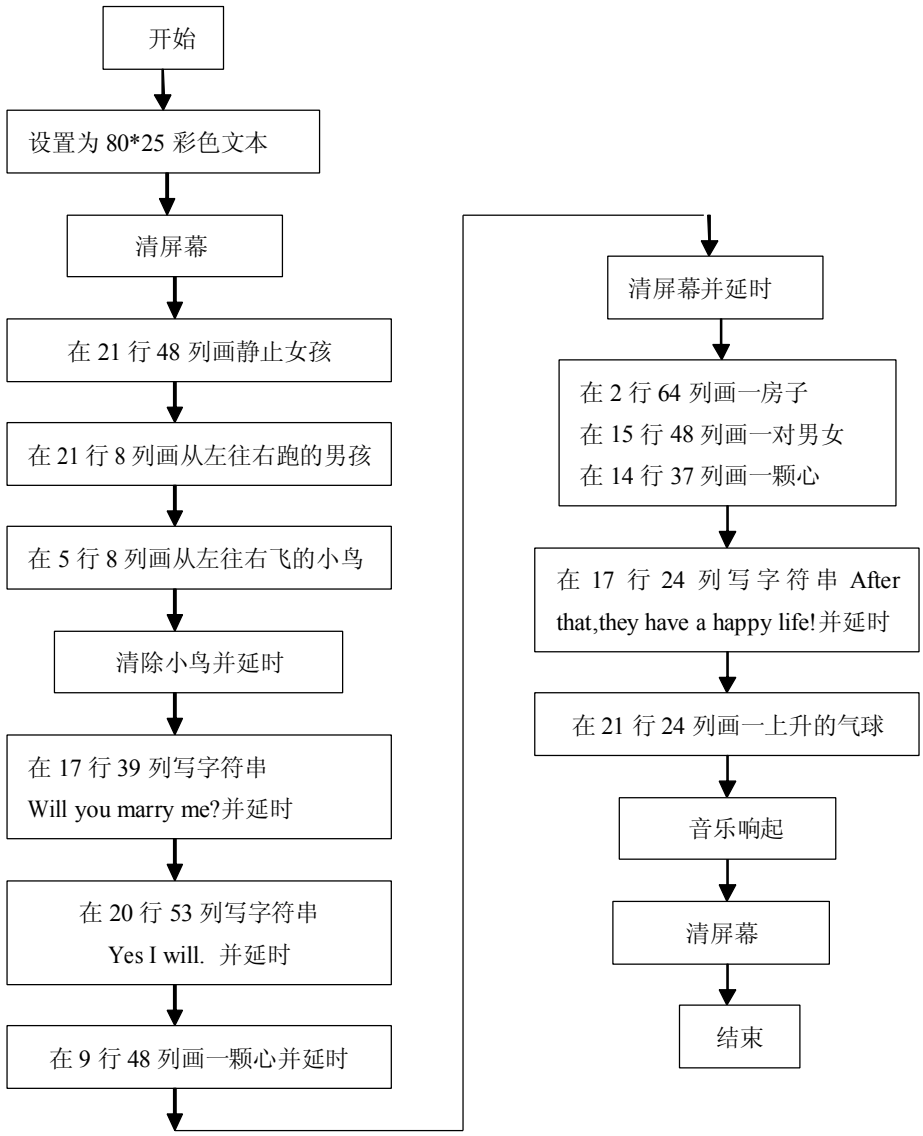

开始
设置为 80*25 彩色文本
清屏幕
在 21 行 48 列画静止女孩
在 21 行 8 列画从左往右跑的男孩
在 5 行 8 列画从左往右飞的小鸟
清除小鸟并延时
在 17 行 39 列写字符串
Will you marry me?并延时
在 20 行 53 列写字符串
Yes I will. 并延时
在 9 行 48 列画一颗心并延时
清屏幕并延时
在 2 行 64 列画一房子
在 15 行 48 列画一对男女
在 14 行 37 列画一颗心
在 17 行 24 列写字符串 After that,they have a happy life!并延时
在 21 行 24 列画一上升的气球
音乐响起
清屏幕
结束

图 3.1　程序总体结构图

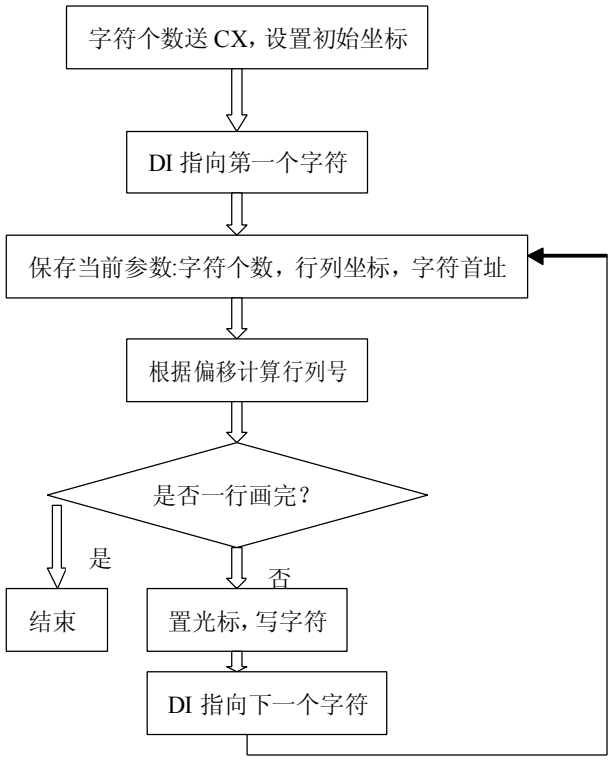

字符个数送 CX，设置初始坐标
DI 指向第一个字符
保存当前参数:字符个数，行列坐标，字符首址
根据偏移计算行列号
是否一行画完？
是
否
结束
置光标，写字符
DI 指向下一个字符

图 3.2　静止程序流程图

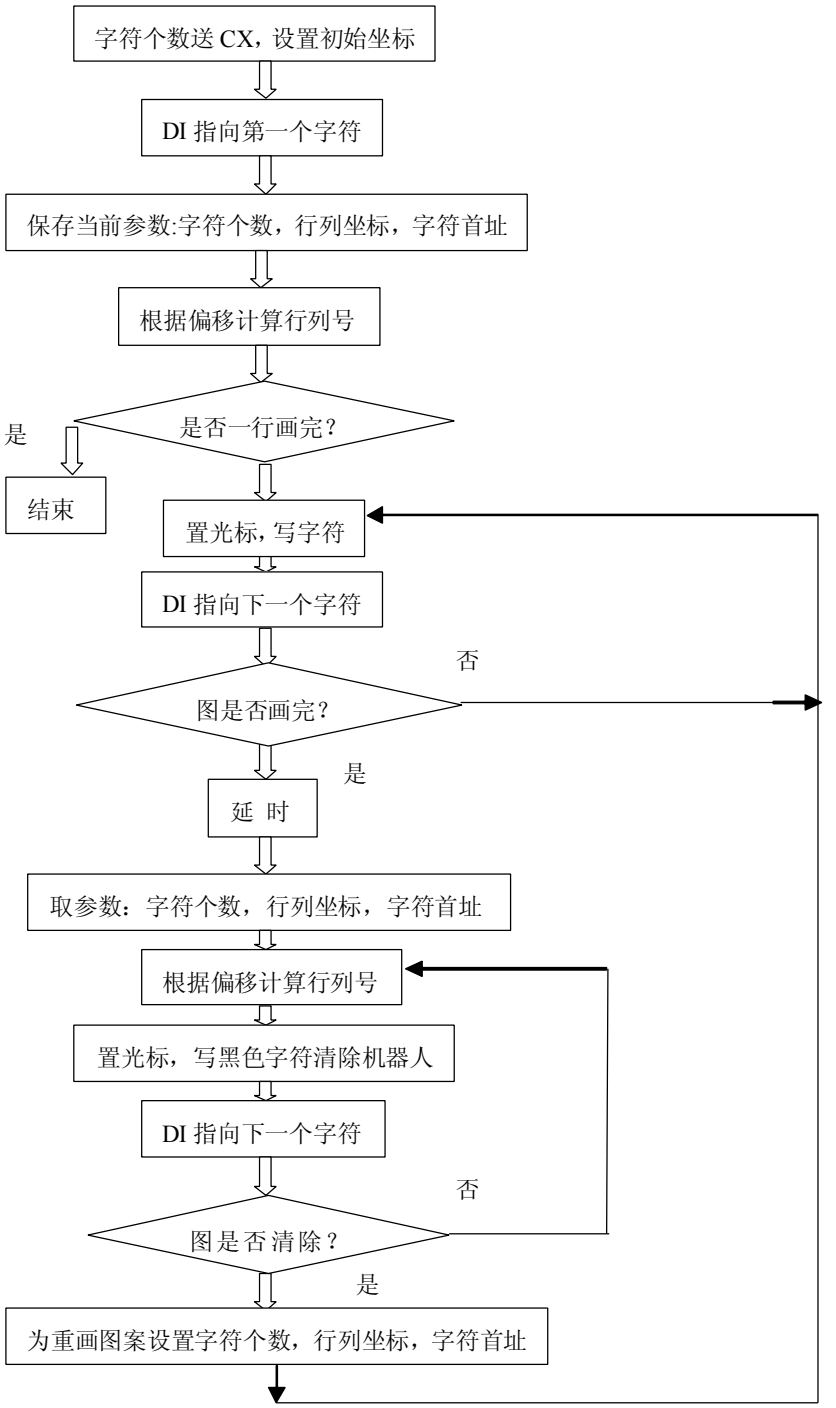
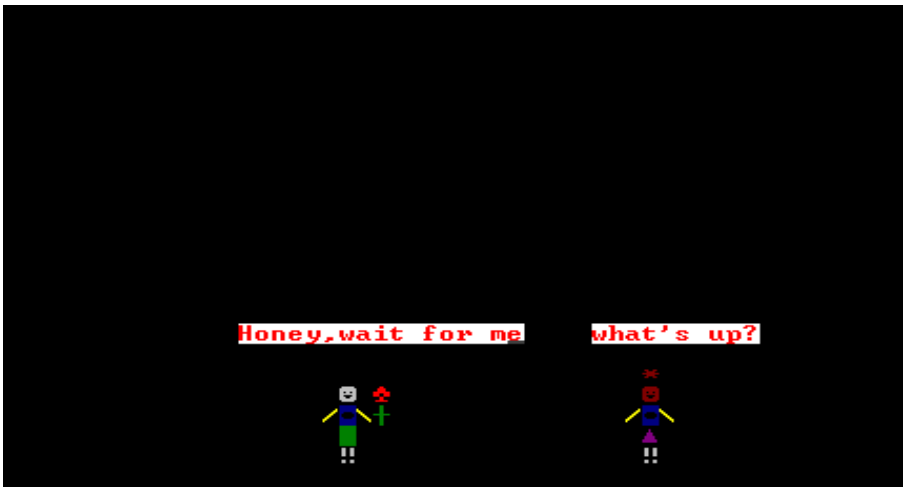
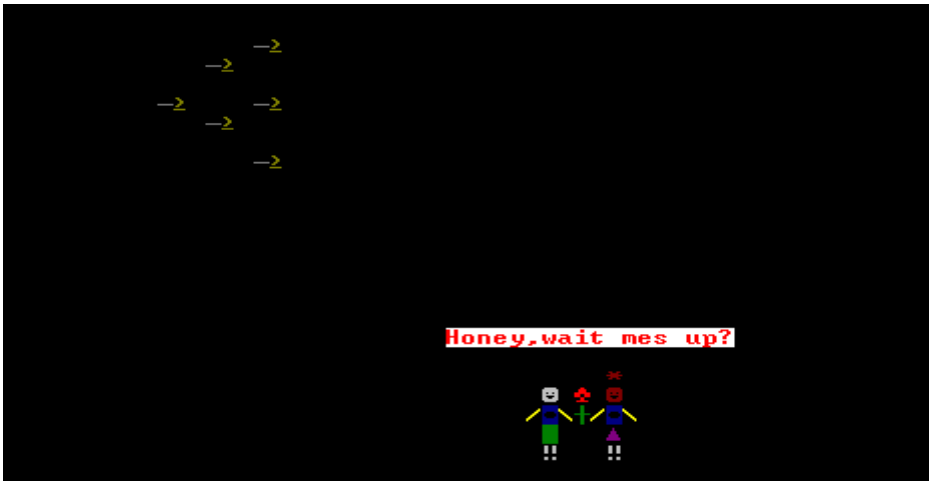


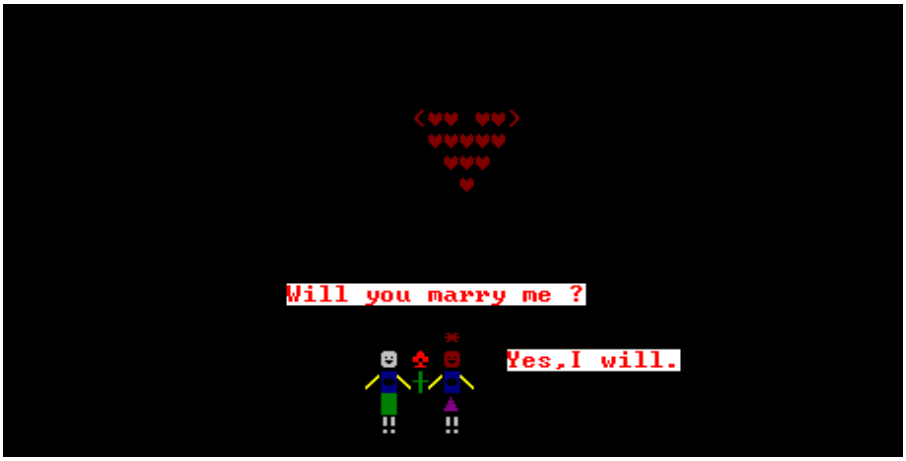
图 3.3　移动程序的流程图



（a） 等待



（b） 相遇，小鸟飞过



（c） 求婚，心形图案降落

（d） 幸福生活开始

（e） 音乐响起，音符往上升起

图 3.4 动画示意图

3.5 设计源程序

```
PUSHR   MACRO X, Y, Z, W
PUSH    X
PUSH    Y
PUSH    Z
PUSH    W
ENDM
；堆栈顶数据弹出到寄存器宏定义
```

```
POPR    MACRO X, Y, Z, W
POP     W
POP     Z
POP     Y
POP     X
ENDM
STACKS      SEGMENT
    DW      256  DUP (?)
STACKS      ENDS
DATA        SEGMENT
ROBOT  DB  8                    ；女孩
       DB  2FH, 0EH, 0, 0       ；左手，字符属性，行偏移，列偏移
       DB  08H, 01H, 0, 1       ；胸部，字符属性，行偏移，列偏移
       DB  5CH, 0EH, 0, 1       ；右手，字符属性，行偏移，列偏移
       DB  1EH, 05H, 1, -1      ；腹部，字符属性，行偏移，列偏移
       DB  13H, 7H, 1, 0        ；两腿，字符属性，行偏移，列偏移
       DB  02H, 04H, -3, 0      ；面部，字符属性，行偏移，列偏移
       DB  2AH, 04H, -1, 0      ；天线，字符属性，行偏移，列偏移
       DB  77H, 0FCH, -2, -3    ；字母 w
       DB  68H, 0FCH, 0, 1      ；字母 h
       DB  61H, 0FCH, 0, 1      ；字母 a
       DB  74H, 0FCH, 0, 1      ；字母 t
       DB  27H, 0FCH, 0, 1      ；标点下引号
       DB  73H, 0FCH, 0, 1      ；字母 s
       DB  0H, 0FCH, 0, 1       ；空格
       DB  75H, 0FCH, 0, 1      ；字母 u
       DB  70H, 0FCH, 0, 1      ；字母 p
       DB  3FH, 0FCH, 0, 1      ；标点问号
       ROBOT1 DB  22            ；男孩
       DB  2FH, 0EH, 0, 0       ；左手，字符属性，行偏移，列偏移
```

```
DB  08H, 01H, 0, 1          ；胸部，字符属性，行偏移，列偏移
DB  5CH, 0EH, 0, 1          ；右手，字符属性，行偏移，列偏移
DB  0C5H, 02H, 0, 1         ；花柄
DB  05H, 0CH, -1, 0         ；花
DB  0DBH, 02H, 2, -2        ；腹部，字符属性，行偏移，列偏移
DB  13H, 7H, 1, 0           ；两腿，字符属性，行偏移，列偏移
DB  02H, 07H, -3, 0         ；面部，字符属性，行偏移，列偏移
DB  48H, 0FCH, -3, -6       ；字母 H
DB  6FH, 0FCH, 0, 1         ；字母 o
DB  6EH, 0FCH, 0, 1         ；字母 n
DB  65H, 0FCH, 0, 1         ；字母 e
DB  79H, 0FCH, 0, 1         ；字母 y
DB  2CH, 0FCH, 0, 1         ；标点逗号
DB  77H, 0FCH, 0, 1         ；字母 w
DB  61H, 0FCH, 0, 1         ；字母 a
DB  69H, 0FCH, 0, 1         ；字母 i
DB  74H, 0FCH, 0, 1         ；字母 t
DB  00H, 0FCH, 0, 1         ；空格
DB  66H, 0FCH, 0, 1         ；字母 f
DB  6FH, 0FCH, 0, 1         ；字母 o
DB  72H, 0FCH, 0, 1         ；字母 r
DB  00H, 0FCH, 0, 1         ；空格
DB  6DH, 0FCH, 0, 1         ；字母 m
DB  65H, 0FCH, 0, 1         ；字母 e
BUFDB  12                   ；小鸟
DB  0C4H, 08H, 0, 0
DB  0F2H, 06H, 0, 1
DB  0C4H, 08H, -2, 2
DB  0F2H, 06H, 0, 1
DB  0C4H, 08H, 3, -1
```

```
        DB  0F2H, 06H, 0, 1
        DB  0C4H, 08H, -4, 2
        DB  0F2H, 06H, 0, 1
        DB  0C4H, 08H, 3, -1
        DB  0F2H，06H, 0, 1
        DB  0C4H, 08H, 3, -1
        DB  0F2H, 06H, 0, 1
BUF1    DB  'Will you marry me ?'
        LEN1=$-BUF1
BUF2    DB  'Yes，I will.'
        LEN2=$-BUF2
HOUSE   DB  90                  ；房子
        DB  2FH, 03H, 0, 0      ；屋顶上升面
        DB  2FH, 03H, 1, -1
        DB  2FH, 03H, 1, -1
        DB  2FH, 03H, 1, -1
        DB  2FH, 03H, 1, -1
        DB  2FH, 03H, 1, -1
        DB  0FEH, 0FH, -2, 5    ；窗户
        DB  0FEH, 0FH, 0, 1
        DB  0FEH, 0FH, 1, -1
        DB  0FEH, 0FH, 0, 1
        DB  5CH, 03H, -4, 0     ；屋顶的下降面
        DB  5CH, 03H, 1, 1
        DB  5CH, 03H, 1, 1
        DB  5CH, 03H, 1, 1
        DB  5CH, 03H, 1, 1
        DB  5CH, 03H, 1, 1
        DB  0A9H, 03H, 1, 0     ；屋顶的横
        DB  0A9H, 03H, 0, -1
```

```
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0A9H, 03H, 0, -1
DB  0B3H, 03H, 1, -1        ；房子的一竖
DB  0B3H, 03H, 1, 0
DB  0B3H, 03H, 1, 0
DB  0B3H, 03H, 1, 0
DB  0B3H, 03H, 1, 0
DB  0B3H, 03H, 1, 0
DB  0B3H, 03H, 1, 0
DB  0DBH, 0EH, 0, 6         ；门
DB  08H, 0E1H, -1, 0
DB  0DBH, 0EH, -1, 0
DB  0DBH, 0EH, 0, 1
DB  0DBH, 0EH, 1, 0
DB  0DBH, 0EH, 1, 0
DB  0B3H, 03H, 0, 6         ；房子的另一竖
DB  0B3H, 03H, -1, 0
DB  0B3H, 03H, -1, 0
DB  0B3H, 03H, -1, 0
DB  0B3H, 03H, -1, 0
DB  0B3H, 03H, -1, 0
DB  0B3H, 03H, -1, 0
```

```
DB  0C4H, 03H, 7, 0          ；房底的横
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0，-1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C4H, 03H, 0, -1
DB  0C5H, 02H, 0, -5         ；左边的花
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 15         ；右边的花
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
DB  0C5H, 02H, 1, 1
DB  05H, 05H, -1, 0
```

```
        DB  0C5H, 02H, 1, 1
        DB  05H, 05H, -1, 0
        DB  0C5H, 02H, 1, 1
        DB  05H, 05H, -1, 0
        DB  0DBH, 08H, 2, -12       ；台阶
        DB  0DBH, 08H, 0, 1
        DB  0DBH, 08H, 2, -3
        DB  0DBH, 08H, 0, 1
        DB  0DBH, 08H, 2, -3
        DB  0DBH, 08H, 0, 1
        DB  0DBH, 08H, 2, -3
        DB  0DBH, 08H, 0, 1
PEOPLE  DB  15                      ；一对男女
        DB  2FH, 0EH, 0, 0          ；左手，字符属性，行偏移，列偏移
        DB  08H, 06H, 0, 1          ；胸部，字符属性，行偏移，列偏移
        DB  5CH, 0EH, 0, 1          ；右手，字符属性，行偏移，列偏移
        DB  1EH, 04H, 1, -1         ；腹部，字符属性，行偏移，列偏移
        DB  13H, 7H, 1, 0           ；两腿，字符属性，行偏移，列偏移
        DB  02H, 04H, -3, 0         ；面部，字符属性，行偏移，列偏移
        DB  2AH, 04H, -1, 0         ；天线，字符属性，行偏移，列偏移
        DB  0C5H, 02H, 2, -2        ；花柄
        DB  05H, 0CH, -1, 0         ；花
        DB  2FH, 0EH, 1, 4          ；左手，字符属性，行偏移，列偏移
        DB  08H, 06H, 0, 1          ；胸部，字符属性，行偏移，列偏移
        DB  5CH, 0EH, 0, 1          ；右手，字符属性，行偏移，列偏移
        DB  0DBH, 0AH, 1, -1        ；腹部，字符属性，行偏移，列偏移
        DB  13H, 7H, 1, 0           ；两腿，字符属性，行偏移，列偏移
        DB  02H, 07H, -3, 0         ；面部，字符属性，行偏移，列偏移
HEART   DB  17
        DB  03H, 04H, 0, 0          ；心
```

```
        DB  03H, 04H, 1, 0
        DB  03H, 04H, -1, 1
        DB  03H, 04H, 1, 0
        DB  03H, 04H, 1, 0
        DB  03H, 04H, -1, 1
        DB  03H, 04H, 1, 0
        DB  03H, 04H, 1, 0
        DB  03H, 04H, -3, 1
        DB  03H, 04H, 1, 0
        DB  03H, 04H, 1, 0
        DB  03H, 04H, -2, 1
        DB  03H, 04H, 1, 0
        DB  29H, 04H, -1, 1
        DB  28H, 04H, 0, -6
ZI      DB  'After that，they have a happy life!'
        LENZI=$-ZI
QIQIU   DB  10
        DB  4FH, 01H, 0, 0                  ；气球
        DB  0F5H, 01H, 1, 0
        DB  4FH, 0BH, -2, 3
        DB  0F5H, 0BH, 1, 0
        DB  4FH, 0DH, -2, 3
        DB  0F5H, 0DH, 1, 0
        DB  4FH, 0EH, 4, -4
        DB  0F5H, 0EH, 1, 0
        DB  4FH, 02H, -2, 3
        DB  0F5H, 02H, 1, 0
FREQ    DW 5 DUP (330) , 294, 262           ；音乐频率
        DW 5 DUP (330)
        DW 5 DUP (330) , 349, 392, 294
```

```
        DW 294, 262, 262, -1
TIME    DW 4 DUP (400) , 800, 400, 400      ；音乐节奏时间
        DW 4 DUP (400) , 800
        DW 8 DUP (500)
        DW 500, 500, 1600
COUNT   DW 0                                ；保存字符个数
POINTER DW 0                                ；保存字符首地址
LINE    DB  0                               ；保存行号
COLUMN  DB  0                               ；保存列号
        DATA    ENDS
        CODES   SEGMENT
                ASSUME CS: CODES, DS: DATA, ES: DATA，SS: STACKS
        START:  MOV     AX, DATA
                MOV     DS, AX
                MOV     ES, AX
                MOV     AH, 0               ；设置为 80×25 彩色文本方式
                MOV     AL, 3
                INT     10H
                CALL    CLEAR               ；清屏幕
                LEA     DI, ROBOT           ；将 ROBOT 数据段首地址送 DI
                MOV     DX, 1530H           ；显示的起始行号和列号
                MOV     BH, 0
                CALL    STOP                ；调用静止子程序
                LEA     DI, ROBOT1          ；将 ROBOT1 数据段首地址送 DI
                MOV     DX, 1508H           ；显示的起始行号和列号
                MOV     BH, 0
                CALL    MOVES               ；调用左到右移动子程序
                LEA     DI, BUF             ；将 BUF 数据段首地址送 DI
                MOV     DX, 0508H           ；显示的起始行号和列号
                MOV     BH, 0
```

```
MOV     SI, 0           ；用来判断调用左到右清除子程序
CALL     MOVES          ；调用左到右移动子程序
CALL     ERASE          ；调用右到左清除子程序
CALL     DELAY1         ；调用延时时间长的子程序
LEA      BP, BUF1       ；将字符串数据段首地址送 BP
MOV     CX, LEN1
MOV     DX, 1127H       ；显示的起始行号和列号
MOV     BH, 0
MOV     BL, 0FCH        ；闪烁的白底红字
MOV     AL, 0
MOV     AH, 13H
INT     10H
CALL    DELAY1          ；调用延时时间长的子程序
LEA     BP, BUF2        ；将字符串数据段首地址送 BP
MOV     CX, LEN2
MOV     DX, 1435H       ；显示的起始行号和列号
MOV     BH, 0
MOV     BL, 0FCH        ；闪烁的白底红字
MOV     AL，0
MOV     AH，13H
INT     10H
CALL    DELAY1          ；调用延时时间长的子程序
LEA     DI, HEART       ；将心数据段首地址送 DI
MOV     DX, 0930H       ；显示的起始行号和列号
MOV     BH, 0
CALL    STOP            ；调用静止子程序
CALL    DELAY1          ；调用延时时间长的子程序
CALL    CLEAR           ；清除屏幕
CALL    DELAY           ；调用延时时间短的子程序
LEA     DI, HOUSE       ；将房子数据段首地址送 DI
```

```
        MOV     DX, 0240H           ；显示的起始行号和列号
        MOV     BH, 0
        CALL    STOP                ；调用静止子程序
        LEA     DI, PEOPLE          ；将人数据段首地址送 DI
        MOV     X, 1530H            ；显示的起始行号和列号
        MOV     BH, 0
        CALL    STOP                ；调用静止子程序
        LEA     DI, HEART           ；将心数据段首地址送 DI
        MOV     DX, 1425H           ；显示的起始行号和列号
        MOV     BH, 0
        CALL    STOP                ；调用静止子程序
        LEA     BP, ZI              ；将字符串数据段首地址送 BP
        MOV     CX, LENZI
        MOV     DX, 1118H           ；显示的起始行号和列号
        MOV     BH, 0
        MOV     BL, 0FCH            ；闪烁的白底红字
        MOV     AL, 0
        MOV     AH, 13H
        INT     10H
        CALL    DELAY1              ；调用延时时间长的子程序
        LEA     DI, QIQIU           ；将气球数据段首地址送 DI
        MOV     DX, 1508H           ；显示的起始行号和列号
        MOV     BH, 0
        MOV     SI，1               ；用来判断调用左上移清除子程序
        CALL    MOVES               ；调用 MOVES 子程序
        MOV     SI, OFFSET FREQ ；将音乐频率数据段首地址送 DI
        MOV     DI, OFFSET TIME ；将音乐节奏时间数据段首地址送 BX
TT:     MOV     CX, [SI]
        CMP     CX, -1
        JE      CC                  ；跳出音乐程序
```

```
        MOV     BX, [DI]
        CALL    GENSOUND
        ADD     SI, 2
        ADD     DI，2
        JMP     TT
        CALL    DELAY1          ；调用延时时间长的子程序
        CALL    CLEAR           ；清除屏幕
    CC: MOV     AH, 4CH         ；主程序结束返回 DOS
        INT     21H
MOVES   PROC                    ；从左往右移动子程序
        PUSHR   AX, BX, CX, DX  ；保存寄存器内容
        PUSH    DI
        XOR     CH, CH
        MOV     CL, [DI]        ；字符个数送 CX
        INC     DI              ；指向第一个显示符号
        MOV     COUNT, CX       ；保存字符个数
        MOV     POINTER, DI     ；保存字符首地址
        MOV     LINE, DH        ；保存行号
        MOV     COLUMN, DL      ；保存列号
NEXT:   ADD     DH, [DI+2]      ；根据偏移值计算下一个符号的行号
        ADD     DL, [DI+3]      ；计算列号
        MOV     AH, 2
        INT     10H             ；设置光标位置
        MOV     AL, [DI]        ；取字符
        MOV     BL, [DI+1]      ；取字符属性
        PUSH    CX              ；保存计数值
        ；设置写彩色字符
        MOV     AH, 9
        MOV     CX, 1
        INT     10H
```

```
        POP     CX              ；恢复计数值
        ADD     DI, 4           ；指向下一个显示字符
        LOOP    NEXT            ；机器人没画完转 NEXT
        CALL    DELAY           ；延时子程序
        CMP     DL, 50
        JA      LL              ；判断是否到 50 列
        CMP     SI0             ；判断调用哪个清除子程序
        JE      AA
        CALL    ERASE1          ；调用左上移清除子程序
        JMP     BB
    AA: CALL    ERASE           ；调用左到右清除子程序
    BB: JMP     SHORT NEXT
    LL: POP     DI
        POPR    AX, BX, CX, DX
        RET
MOVES   ENDP
ERASE   PROC                    ；清除子程序
        MOV     CX, COUNT       ；字符个数送 CX
        MOV     DI, POINTER     ；字符首地址送 DI
        MOV     DH, LINE        ；行号送 DH
        MOV     DL, COLUMN      ；列号送 DL
    L:  ADD     DH, [DI+2]      ；根据相对偏移计算行号
        MOV     AH, 2           ；设置光标位置
        ADD     DL, [DI+3]
        INT     10H
        MOV     AL, [DI]        ；取字符
        MOV     BL, 0           ；字符属性为黑底黑字
        PUSH    CX
        MOV     AH, 9           ；写字符
        MOV     X, 1
```

```
        INT     10H
        POP     CX
        ADD     DI, 4           ；指向下一个字符
        LOOP    L               ；未清除完转 L
        MOV     CX, COUNT       ；为重画机器人做准备
        MOV     DI, POINTER
        MOV     DH, LINE
        INC     COLUMN
        MOV     DL, COLUMN
        RET
ERASE   ENDP
ERASE1  PROC                    ；清楚左上移子程序
        MOV     CX, COUNT       ；字符个数送 CX
        MOV     DI, POINTER     ；字符首地址送 DI
        MOV     DH, LINE        ；行号送 DH
        MOV     DL, COLUMN      ；列号送 DL
    L2: ADD     DH, [DI+2]      ；根据相对偏移计算行号
        MOV     AH, 2           ；设置光标位置
        ADD     DL, [DI+3]
        INT     10H
        MOV     AL, [DI]        ；取字符
        MOV     BL, 0           ；字符属性为黑底黑字
        PUSH    CX
        MOV     AH, 9           ；写字符
        MOV     CX, 1
        INT     10H
        POP     CX
        ADD     DI, 4           ；指向下一个字符
        LOOP    L2              ；未清除完转 L2
        MOV     CX, COUNT       ；为重画做准备
```

```
        MOV     DI, POINTER
        DEC     LINE            ；上移
        MOV     DH, LINE
        INC     COLUMN
        MOV     DL, COLUMN
        RET
ERASE1  ENDP
DELAY   PROC                    ；延时时间短子程序
        PUSHR   AX, BX, CX, DX
        MOV     DX, 9000
GO:     MOV     CX, 8000
REPEAT: LOOP    REPEAT
        DEC     DX
        JNE     GO
        POPR    AX, BX, CX, DX
        RET
DELAY   ENDP
DELAY1  PROC                    ；延时子时间长程序
        PUSHR   AX, BX, CX, DX
        MOV     DX, 9000H
GO1:    MOV     CX, 9000H
REPE1:  LOOP    REPE1
        DEC     DX
        JNE     GO1
        POPR    AX，BX，CX，DX
        RET
DELAY1  ENDP
CLEAR   PROC                    ；清屏幕子程序
            PUSHR   AX, BX, CX, DX
            MOV     BH, 7
```

```
                MOV     DX, 184FH
                MOV     AX, 0600H
                MOV     CX, 0
                INT     10H
                POPR    AX, BX, CX, DX
                RET
CLEAR           ENDP
STOP            PROC                    ；静止子程序
                PUSHR   AX, BX, CX, DX  ；保存寄存器内容
                XOR     CH, CH
                MOV     CL, [DI]        ；字符个数送 CX
                INC     DI              ；指向第一个显示符号
                MOV     COUNT, CX       ；保存字符个数
                MOV     POINTER, DI     ；保存字符首地址
                MOV     LINE, DH        ；保存行号
                MOV     COLUMN, DL      ；保存列号
NEXT1:          ADD     DH, [DI+2]      ；根据偏移值计算下一个符号
                                        ；的行号
                ADD     DL, [DI+3]      ；计算列号
                MOV     AH, 2
                INT     10H             ；设置光标位置
                MOV     AL, [DI]        ；取字符
                MOV     BL, [DI+1]      ；取字
                PUSH    CX
        ；设置写彩色字符
                MOV     AH, 9
                MOV     CX, 1
                INT     10H
                POP     CX
                ADD     DI, 4           ；指向下一个显示字符
```

```
                LOOP    NEXT1                   ；没画完转 NEXT1
                POPR    AX, BX, CX, DX
                RET
STOP    ENDP
GENSOUND PROC    NEAR
                PUSH    DX
                MOV     AL, 0B6H
                OUT     43H, AL
                MOV     DX, 08H
                MOV     AX, 3208H
                DIV     CX
                OUT     42H, AL
                MOV     AL, AH
                OUT     42H, AL
                IN      AL, 61H
                MOV     AH, AL
                OR      AL, 3
                OUT     61H, AL
        S2:     PUSH    DX
                PUSH    AX
                MOV     DX, 8H
                MOV     AX, 0F05H
        S1:     SUB     AX, 1
                SBB     DX, 0
                JNZ     S1
                POP     AX
                POP     DX
                DEC     BX
                JNZ     S2
                MOV     AL, AH
```

```
                OUT     61H, AL
                POP     DX
                RET
GENSOUND ENDP
CODES           ENDS
                END     START
```

4. 问题讨论与分析

4.1 如何实现图画？

```
ROBOT   DB  18                  ；女孩
        DB  2FH, 0EH, 0, 0      ；左手，字符属性，行偏移，列偏移
        DB  08H, 01H, 0, 1      ；胸部，字符属性，行偏移，列偏移
        DB  5CH, 0EH, 0, 1      ；右手，字符属性，行偏移，列偏移
        DB  1EH, 05H, 1, -1     ；腹部，字符属性，行偏移，列偏移
        DB  13H, 7H, 1, 0       ；两腿，字符属性，行偏移，列偏移
        DB  02H, 04H, -3, 0     ；面部，字符属性，行偏移，列偏移
        DB  2AH, 04H, -1, 0     ；天线，字符属性，行偏移，列偏移
```

4.2 如何实现静止的图画？

```
        PUSHR   AX, BX, CX, DX  ；保存寄存器内容
        XOR     CH, CH
        MOV     CL, [DI]        ；字符个数送 CX
        INC     DI              ；指向第一个显示符号
        MOV     COUNT, CX       ；保存字符个数
        MOV     POINTER, DI     ；保存字符首地址
        MOV     LINE, DH        ；保存行号
        MOV     COLUMN, DL      ；保存列号
NEXT1:  ADD     DH, [DI+2]      ；根据偏移值计算下一个符号的行号
        ADD     DL, [DI+3]      ；计算列号
        MOV     AH, 2
        INT     10H             ；设置光标位置
        MOV     AL, [DI]        ；取字符
```

```
        MOV     BL, [DI+1]          ；取字
        PUSH    CX
        ；设置写彩色字符
        MOV     AH, 9
        MOV     CX, 1
        INT     10H
        POP     CX
        ADD     DI, 4               ；指向下一个显示字符
        LOOP    NEXT1               ；没画完转 NEXT1
        POPR    AX, BX, CX, DX
```

以上代码主要运用到单重循环结构。

4.3 如何实现移动的图画？在静止的基础上，运用清除程序，也就是把图画涂成和屏幕背景一样的颜色，看似被清除，实际上没有，因为视觉原因。

```
        MOV     CX, COUNT           ；字符个数送 CX
        MOV     DI, POINTER         ；字符首地址送 DI
        MOV     DH, LINE            ；行号送 DH
        MOV     DL, COLUMN          ；列号送 DL
L:      ADD     DH, [DI+2]          ；根据相对偏移计算行号
        MOV     AH, 2               ；设置光标位置
        ADD     DL, [DI+3]
        INT     10H
        MOV     AL, [DI]            ；取字符
        MOV     BL, 0               ；字符属性为黑底黑字
        PUSH    CX
        MOV     AH, 9               ；写字符
        MOV     CX, 1
        INT     10H
        POP     CX
        ADD     DI, 4               ；指向下一个字符
        LOOP    L                   ；未清除完转 L
```

```
MOV     CX, COUNT       ；为重画机器人做准备
MOV     DI, POINTER
MOV     DH, LINE
INC     COLUMN
MOV     DL, COLUMN
```

实例 2　动画设计“我爱大自然”

1. 设计要求

1）了解并掌握汇编语言设计的一般方法，具备初步的独立分析和设计能力；

2）初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

3）提高综合运用所学的理论知识和方法独立分析和解决问题的能力。

2. 主要仪器设备（实验用的软硬件环境）

硬件环境：PC 机

软件环境：DOS 操作系统

3. 设计内容

3.1 设计内容及功能实现

这个程序的设计是调用 BIOS 中断子程序，在 80*25 彩色文本方式编写一个能移动的小汽车程序，并根据通用发声系统工作原理和乐理知识编写了一段音乐程序。

整个程序实现的效果：开始时，首先在屏幕上显示“I LOVE NATURE，LET US GO AIRING”，让读者一开始就知道我这个程序的目的。一群小鸟在天空中悠哉游哉地飞往一个小树林，随后，一辆小汽车在两旁种着绿树的林荫大道上不无悠闲地享受着大自然的一草一木，一风一物，不等小汽车走后，音乐突然响起，一首《大海》给了在大自然游玩的游客另一番精神享受。整个程序显示了游客在外兜风的一幅悠闲景象。

3.2 设计思路分析

这个程序包含比较多的景物，既有静态的也有动态的，其中还有一段音乐。为了节省存储空间，提高程序设计的效率和质量，使程序简洁、清晰，便于读者阅读，同时也为了便于修改和扩充，采用子程序设计技术和宏定义，根据程序要实现的若干主要功能及个功能块要调用的公共部分，将程序划分为若干个相对独立的模块，为每一模块编制独立的程序段，最后将这些子程序根据调用关系连成一个整体。

这样，整个程序就被分为几个子程序的有机统一。根据 BIOS 中断调用原理，设置 80*25 彩色文本显示方式，分别编写一个子程序显示“I LOVE NATURE，LET US GO AIRING”和一个子程序在屏幕上“画”树。这两个子程序所体现出来的事物都是静止的。为了实现小鸟

能在空中翩翩飞舞和小汽车在林荫路上行驶，还要调用擦除子程序和延时子程序。擦除子程序的原理是根据相对位移设置光标位置，将原来字符属性设置为黑底黑字，以达到擦除效果。延时子程序的原理也很简单，即设置循环次数，让 CPU 做一些“无用功”，这样读者就能看到小鸟“飞起来”和小汽车“跑起来”。可以改变循环次数来控制延时，达到读者想要的效果。动态的事物就这样产生了。

3.3 发声原理和编程方法

PC 机上的大多数输入/输出（I/O）都是由系统插件板上的 8255(或 8255A)可编程外围接口芯片(PPI)管理的。PPI 包括三个 8 位寄存器，两个用于输入功能，一个用于输出功能。输入寄存器分配的 I/O 端口号为 60H 和 62H，输出寄存器分配的 I/O 端口号为 61H。当输出寄存器（I/O 端口 61H）的第 0 位为 1 时，控制 8254 定时器来驱动扬声器，当第 1 位为 1 时，扬声器的门电路接通，并一直保持到位 1 变为 0 时关闭。即控制电路能以位触发和定时器控制两种不同的方式驱动扬声器发声。计算机中的三个定时器由可编程定时芯片（PROGRAMMABLE INTERVAL TIMER/PIT）实现，定时器 2 用于发声，发声的频率取决于定时器计数器的初始设置值，定时器会将计数值从 初始值开始递减到 0，然后将计数值复原为初始值，以此循环。计数值 的递减由频率为 1，193，180 HZ 的系统振荡器控制着，该频率值在所有的 PC 家族的机器中是固定的。振荡器每产生一个脉冲，计数值就减一，为了发出不同频率的声音，我们只需要设定不同的初始值即可，发声频率和初始值的关系是：声音频率=1193180/初始值。

根据乐理知识，在程序中定义两组数据，即频率数据 FREQ 和节拍时间数据 TIME。频率数据可直接用定时器时钟 1193180 HZ/已知频率 F 得到，即 FREQ=1234DCH/F.如果将已知频率 F 保存在 DI，可用以下指令来自动求出 FREQ 的值。

```
MOV     DX, 0012H
MOV     AX, 34DCH
DIV     DI
```

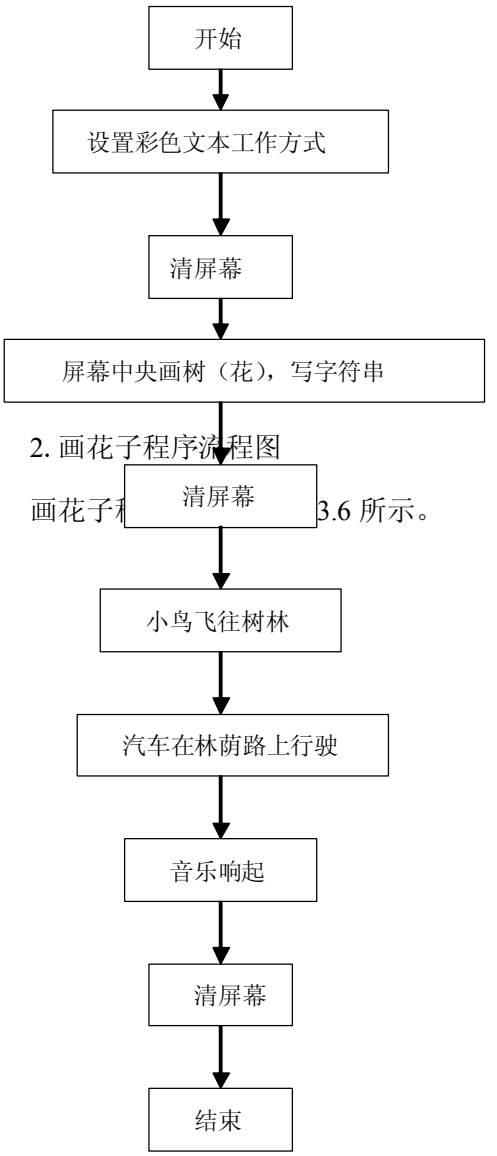
从而产生出方波的记数值。由于乐曲属于人耳听到的频率范围，所以已知频率范围在 19～20000HZ 之间。节拍时间数据 TIME=节拍数*节拍时间。

例如 1=C 调、拍号 4/4 的乐曲中，每 1 拍为 1/4S，2 拍为 1/2S，4 拍为 1S，依次类推。音符 5 的 TIME=2 拍*1/4S=2*0.25=0.5S=500mS=50*10mS。

3.4 程序流程图

1. 程序总流程图

程序总流程图如图 3.5 所示。



2. 画花子程序流程图

画花子程序流程图如图 3.6 所示。

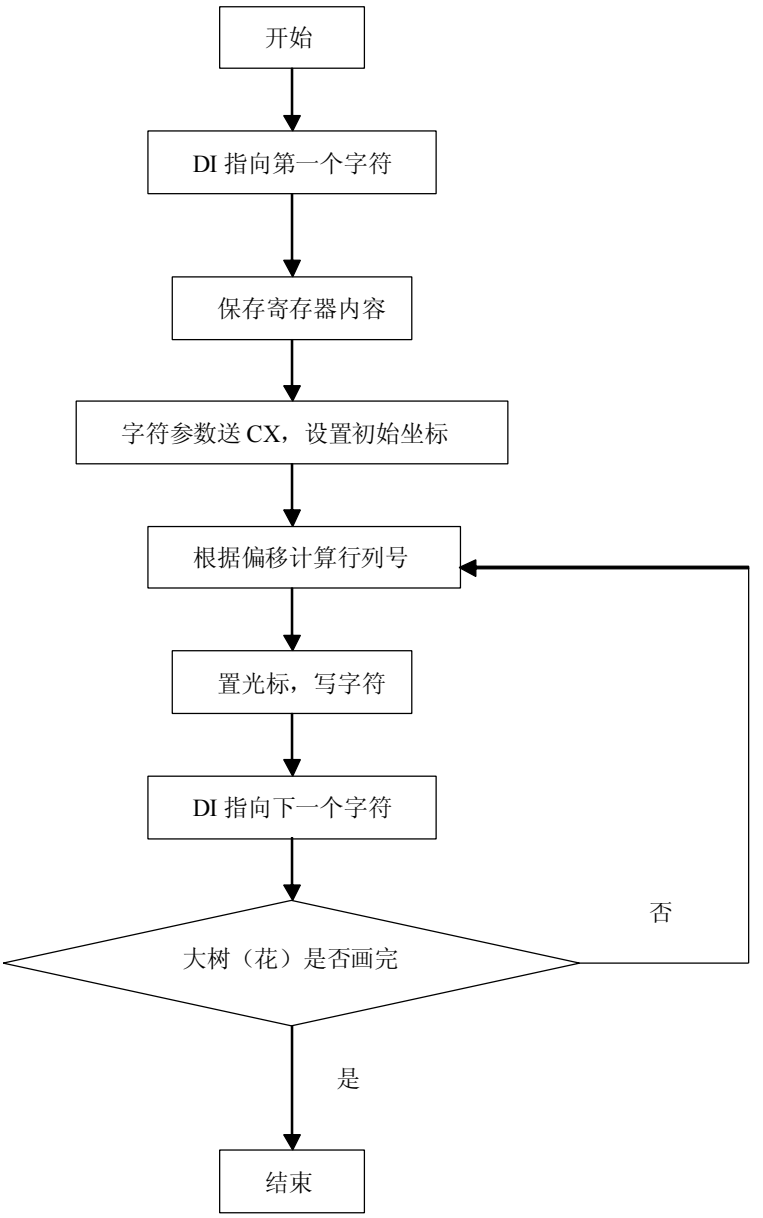


图 3.6　画花子程序流程图

3. 在道路两旁画小树（小草）子程序流程图

道路两旁画小树（小草）子程序流程图见图 3.7 所示。

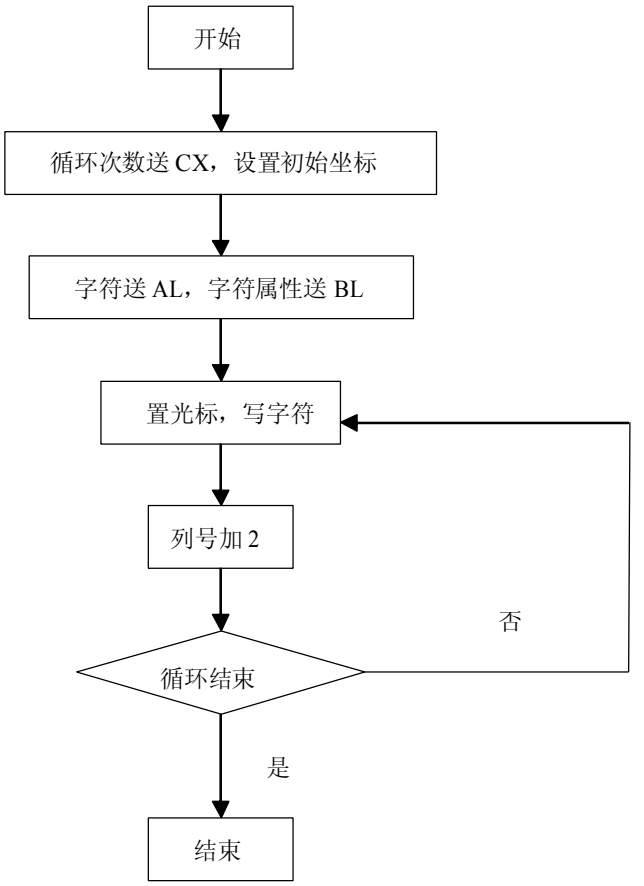


图 3.7　道路两旁画小树（小草）子程序流程图

4. 小鸟飞翔子程序流程图

小鸟飞翔子程序流程图如图 3.8 所示

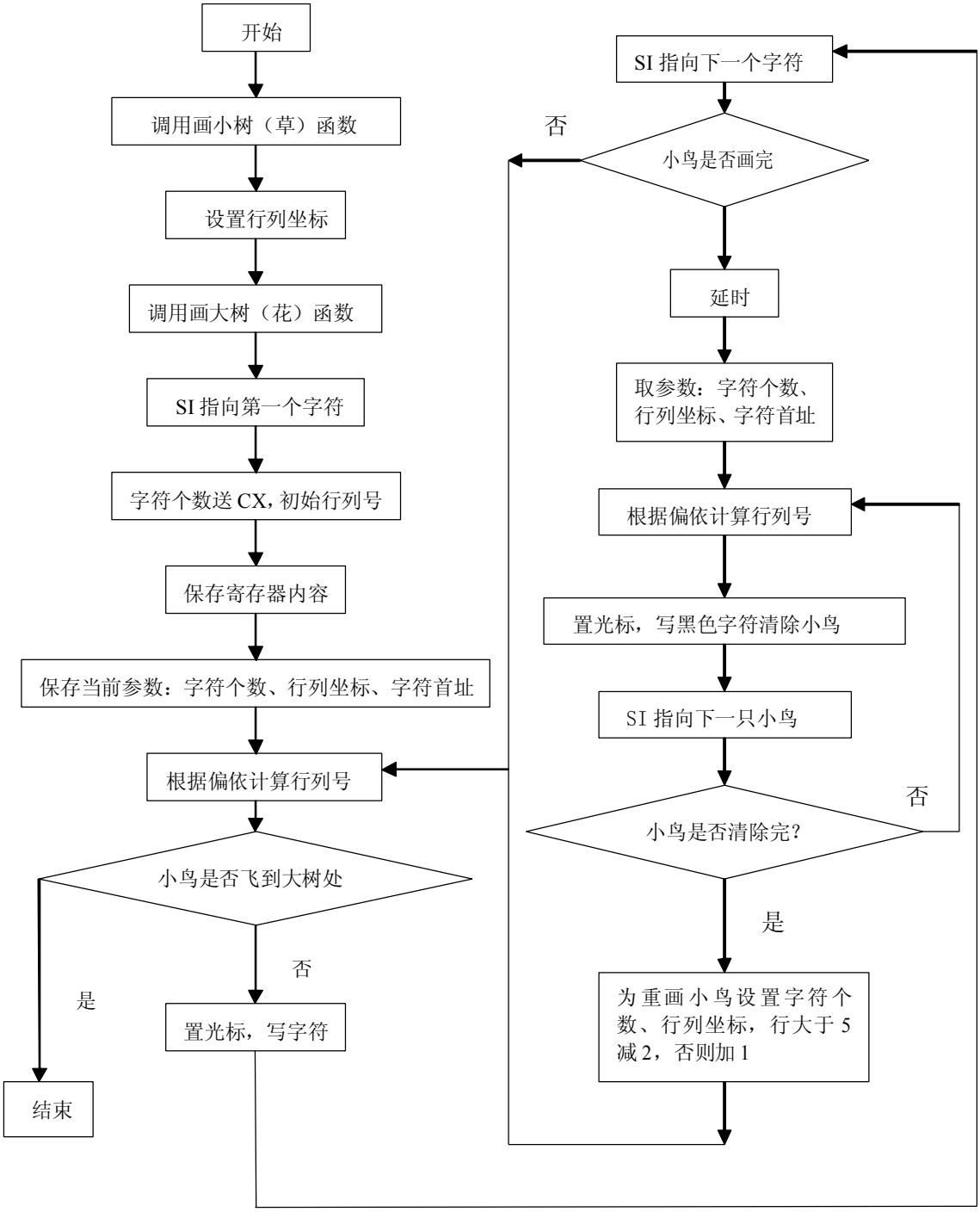
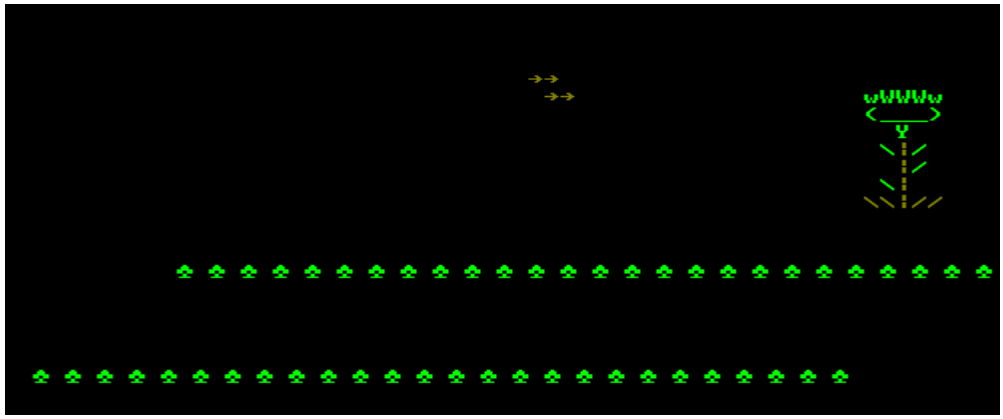


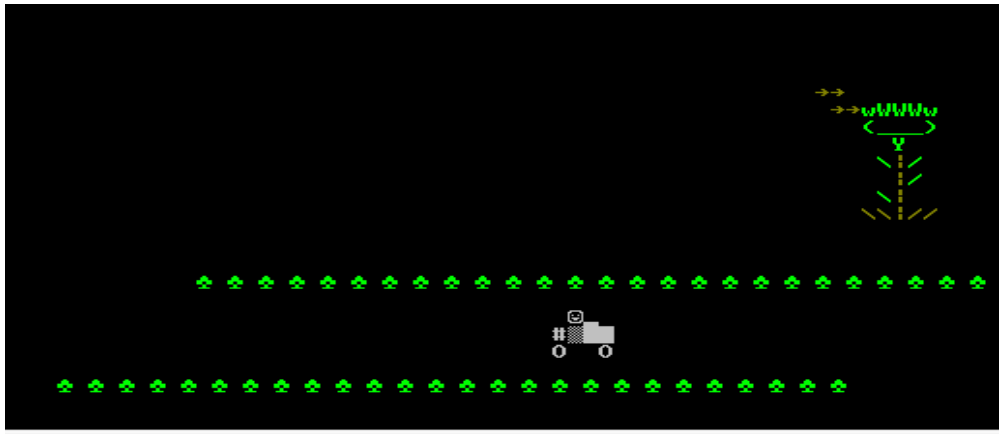
图 3.8 小鸟飞翔子程序流程图

3.5 动画示意图见图 3.9 所示。

图（a） 树林



图（b） 道路两旁小草 小鸟飞过



图（c） 小车开过 音乐响起

图 3.9 动画示意图

3.6 源程序

```
SETIOM    MACRO X, Y, Z, L, M, N          ；功能调用宏定义
          MOV   X, Y
          Z     L, M
          INT   N
          ENDM
PUSHR     MACRO X, Y, Z, W                ；寄存器压入堆栈宏定义
          PUSH  X
          PUSH  Y
          PUSH  Z
          PUSH  W
ENDM
POPR      MACRO X, Y, Z, W                ；栈顶数据弹出到寄存器宏定义
          POP   W
          POP   Z
          POP   Y
          POP   X
ENDM
STACKS    SEGMENT                         ；堆栈段
     DW   256 DUP (?)
STACKS    ENDS
DATA      SEGMENT                         ；数据段
STRING1   DB  'I LOVE NATURE, LET US GO AIRING'
                                          ；要显示的字符串'I LOVE
                                          ；NATURE，LET US GO AIRING'
STRLEN1   EQU   $-STRING1                 ；串长
FLOWER    DB    23                        ；画（大树）字符图形参数表
          DB    'W' , 0AH, 0, 0
          DB    'W' , 0AH, 0, 1
          DB    'W' , 0AH, 0, 1
```

```
        DB      'W', 0AH, 0, 1
        DB      'W', 0AH, 0, 1
        DB      ')', 0AH, 1, 0
        DB      '_', 0AH, 0, -1
        DB      '_', 0AH, 0, -1
        DB      '_', 0AH, 0, -1
        DB      '(', 0AH, 0, -1
        DB      'Y', 0AH, 1, 2
        DB      '|', 06H, 1, 0
        DB      '/', 0AH, 0, 1
        DB      '\', 0AH, 0, -2
        DB      '|', 06H, 1, 1
        DB      '/', 0AH, 0, 1
        DB      '|', 06H, 1, -1
        DB      '\', 0AH, 0, -1
        DB      '|', 06H, 1, 1
        DB      '/', 06H, 0, 1
        DB      '/', 06H, 0, 1
        DB      '\', 06H, 0, -3
        DB      '\', 06H, 0, -1
BIRD    DB      5                       ; 小鸟字符图形参数表
        DB      1AH, 06H, 0, 0
        DB      1AH, 06H, 0, 1
        DB      1AH, 06H, 1, 0
        DB      1AH, 06H, 0, 1
        DB      20H, 00H, 0, 1
CAR     DB      8                       ; 小汽车的字符图形表
        DB      23H, 7, 0, 0
        DB      0B1H, 7, 0, 1
        DB      0DBH, 7, 0, 1
```

```
        DB      0DBH, 7, 0, 1
        DB      4FH, 7, 1, 0
        DB      4FH, 7, 0, -3
        DB      1, 7, -2, 1
        DB      20H, 0, 0, 1
COUNT   DW      0                       ；保存字符个数
POINTER DW      0                       ；保存字符首址
LINE    DB      0                       ；保存行号
COLUMN  DB      0                       ；保存列号
TREE    DB      5, 0AH
FREQ    DW      196, 220                ；定义曲谱音符的频率
        DW      262, 262, 262, 262, 262, 220, 196
        DW      262, 262, 262, 262, 294, 262, 220, 262
        DW      294, 294, 294, 294, 294, 262, 220
        DW      294, 294, 294, 294, 330, 294, 330, 392
        DW      440, 440, 392, 440, 392, 330
        DW      294, 294, 330, 294, 262, 220, 196, 220
        DW      262, 262, 262, 262, 262, 220
        DW      262, 196, 220
        DW      440, 440, 392, 440, 524, 440
        DW      392, 330, 294, 262, 220, 196, 220
        DW      262, 262, 262, 262, 294, 262
        DW      262, 330, 392
        DW      440, 440, 440, 440, 524, 440
        DW      392, 392, 392, 440, 392, 330, 294
        DW      262, 262, 262, 262, 294
        DW      330, 330, 294
        DW      262, 262, 262, 262, 524, 440
        DW      392, 392, 392, 440, 392, 330, 392
        DW      440, 524, 524, 440, 392
```

```
        DW      392，330，392
        DW      440, 440, 440, 440, 524, 440
        DW      392, 392, 392, 440, 392, 330, 294
        DW      262, 262, 262, 262, 392
        DW      330, 330, 294
        DW      262, 262, 262, 262, 294, 330
        DW      392, 392, 330, 392, 330, 392
        DW      440
        DW      9, 9, 196, 660, 294, 294, 262
        DW      262, 1
TIME    DW      400, 400                                    ；定义节拍时间数据
        DW      400, 200, 400, 400, 800, 400, 400
        DW      400, 200, 400, 200, 200, 800, 400, 400
        DW      400, 200, 400, 400, 800, 400, 400
        DW      400, 200, 400, 200, 200, 800, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      400, 200, 200, 400, 400, 800, 400, 400
        DW      400, 200, 400, 400, 800, 800
        DW      1600, 800, 800
        DW      400, 800, 400, 800, 400, 400
        DW      400, 400, 400, 400, 800, 400, 400
        DW      400, 800, 400, 800, 400, 200
        DW      2400, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      400, 800, 200, 200, 800, 400, 400
        DW      400, 800, 400, 800, 800
        DW      2400, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      400, 800, 200, 200, 800, 400, 400
        DW      800, 400, 800, 400, 200
```

```
        DW      2400, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      400, 800, 200, 200, 800, 400, 400
        DW      400, 800, 400, 800, 800
        DW      2400, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      400, 800, 400, 800, 400, 400
        DW      3200
        DW      800, 400, 400, 400, 400, 400, 400
        DW      4000
DATA    ENDS
CODES   SEGMENT
        ASSUME  CS: CODES, DS: DATA, ES: DATA, SS:STACKS
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     ES, AX
        SETIOM  AH, 0, MOV, AL, 3, 10H   ；设置为 80×25 彩色文本方式
        CALL    CLEAR
        CALL    WRITE1                   ；显示'I LOVE NATURE，LET US
                                         ；GO AIRING'
        CALL    DELAY2                   ；较长延迟
        CALL    CLEAR                    ；清屏幕
        CALL    MOVEBIRD                 ；小鸟飞往树林
        CALL    MOVECAR                  ；小汽车在林荫路上行驶
        CALL    MUSIC                    ；音乐响起
        CALL    CLEAR                    ；清屏幕
        MOV     AX, 4C00H
        INT     21H                      ；程序结束
；在屏幕中央画 4 棵大树（花）
；并在大树（花）底下写字符串'I LOVE NATURE, LET US GO AIRING'子程序：
```

```
WRITE1
WRITE1    PROC
          MOV     DX, 0512H              ；定义起始行列号
          CALL    HUA                    ；调用画大树(花)子程序
          MOV     DX, 051BH              ；重新定义起始行列号
          CALL    HUA                    ；再调用画大树(花)子程序
          MOV     DX, 0525H
          CALL    HUA
          MOV     DX, 052EH
          CALL    HUA
          PUSHR   AX, BX, CX, DX         ；保存积存器内容
          LEA     BP, STRING1            ；字符串首地址送 BP
          MOV     CX, STRLEN1            ；串长送 CX
          MOV     DX, 0E13H              ；确定字符串的起始位置
          MOV     BH, 0
          MOV     BL, 84H                ；字符属性()
          SETIOM  AL, 0, MOV, AH, 13H, 10H ；以方式 0 写字符串
          POPR    AX, BX, CX, DX         ；恢复积存器内容
          RET
WRITE1    ENDP
；小鸟飞翔子程序：MOVEBIRD
MOVEBIRD  PROC
          CALL    DRAWTREE               ；调用画小草子程序
          CALL    DRAWTREE1              ；在道路两旁画小草
          MOV     DX, 063CH              ；为大树(花)定义起始行列号
          CALL    HUA                    ；画大树(花)
          LEA     SI, BIRD               ；小鸟字符参数表首地址送 SI
          MOV     DX, 1403H              ；初始行列号
          PUSHR   AX, BX, CX, DX         ；保存寄存器内容
          PUSH    SI
```

```
                XOR     CH, CH
                MOV     CL, [SI]                ；字符个数送 CX
                INC     SI                      ；指向第一个显示符号
                MOV     COUNT, CX               ；保存字符个数
                MOV     POINTER, SI             ；保存字符首地址
                MOV     LINE, DH                ；保存行号
                MOV     COLUMN, DL              ；保存列号
NEXTB:          ADD     DH, [SI+2]              ；根据偏移值算下一个符号的行号
                ADD     DL, [SI+3]              ；计算列号
                CMP     DL, 60                  ；判断小鸟是否到大树
                JB      WRITEB                  ；否，继续画
                POP     SI
                POPR    AX, BX, CX, DX
                RET
WRITEB:         MOV     AH, 2
                INT     10H                     ；设置光标位置
                MOV     AL, [SI]                ；取字符
                MOV     BL, [SI+1]              ；取字符属性
                PUSH    CX                      ；保存计数值
                ；设置写彩色字符
                MOV     AH, 9
                MOV     CX, 1
                INT     10H
                POP     CX                      ；恢复计数值
                ADD     SI, 4                   ；指向下一个显示字符
                LOOP    NEXTB                   ；小鸟没画完转 NEXTB
                CALL    DELAY                   ；调用延时子程序
                CALL    ERASEBIRD               ；调用清除小鸟子程序
                JMP     SHORT NEXTB
MOVEBIRD        ENDP
```

```
；擦除小鸟子程序：ERASEBIRD
ERASEBIRD PROC
          MOV     CX, COUNT              ；字符个数送 CX
          MOV     SI, POINTER            ；字符首地址送 DI
          MOV     DH, LINE               ；行号送 DH
          MOV     DL, COLUMN             ；列号送 DL
    LB:   ADD     DH, [SI+2]             ；根据相对偏移计算行号
          MOV     AH, 2                  ；设置光标位置
          ADD     DL, [SI+3]
          INT     10H
          MOV     AL, [SI]               ；取字符
          MOV     BL, 0                  ；字符属性为黑底黑字
          PUSH    CX
          SETIOM AH, 9, MOV, CX, 1, 10H  ；写字符
          POP     CX
          ADD     SI, 4                  ；指向下一个字符
          LOOP    LB                     ；未清除完转 LB
          MOV     CX, COUNT              ；为重画小鸟做准备
          MOV     SI, POINTER
          CMP     LINE, 5H               ；把小鸟飞翔高度限定在一定范
                                         ；围，更具动态感
          JA      SUBT                   ；大于的时候则跳转 SUBT
          JMP     AD                     ；小于则加 1
    AD:   INC     LINE
          JMP     MOVE
SUBT:     SUB     LINE, 2                ；大于 5 时减 2
MOVE:     MOV     DH, LINE
          INC     COLUMN
          MOV     DL, COLUMN
          RET
```

```
ERASEBIRD       ENDP
；小汽车移动子程序：MOVECAR
MOVECAR  PROC
         LEA     DI, CAR
         MOV     DX, 1300H
         PUSHR   AX, BX, CX, DX
         PUSH    DI
         XOR     CH, CH
         MOV     CL, [DI]                ；字符个数送 CX
         INC     DI                      ；指向第一个显示符号
         MOV     COUNT, CX               ；保存字符个数
         MOV     POINTER, DI             ；保存字符首地址
         MOV     LINE, DH                ；保存行号
         MOV     COLUMN, DL              ；保存列号
NEXTC:   ADD     DH, [DI+2]              ；根据偏移值算下一个符号的行号
         ADD     DL, [DI+3]              ；计算列号
         CMP     DL, 80                  ；判断是否到屏幕最右端
         JB      WRITEC                  ；若否，继续画
         POP     DI
         POPR    AX, BX, CX, DX
         RET
WRITEC:  MOV     AH, 2
         INT     10H                     ；设置光标位置
         MOV     AL, [DI]                ；取字符
         MOV     BL, [DI+1]              ；取字符属性
         PUSH    CX                      ；保存计数值
         ；设置写彩色字符
         MOV     AH, 9
         MOV     CX, 1
         INT     10H
```

```
            POP     CX                  ；恢复计数值
            ADD     DI, 4               ；指向下一个显示字符
            LOOP    NEXTC               ；小汽车没画完转 NEXTC
            CALL    DELAY               ；调用延时子程序
            CALL    ERASECAR            ；调用清除小汽车子程序
            JMP     SHORT NEXTC
MOVECAR     ENDP
；清除小汽车子程序：ERASECAR
ERASECAR    PROC
            MOV     CX, COUNT           ；字符个数送 CX
            MOV     DI, POINTER         ；字符首地址送 DI
            MOV     DH, LINE            ；行号送 DH
            MOV     DL, COLUMN          ；列号送 DL
    LL:     ADD     DH, [DI+2]          ；根据相对偏移计算行号
            MOV     AH, 2               ；设置光标位置
            ADD     DL, [DI+3]
            INT     10H
            MOV     AL, [DI]            ；取字符
            MOV     BL, 0               ；字符属性为黑底黑字
            PUSH    CX
            MOV     AH, 9               ；写字符
            MOV     CX, 1
            INT     10H
            POP     CX
            ADD     DI, 4               ；指向下一个字符
            LOOP    LL                  ；未清除完转 LL
            MOV     CX, COUNT           ；为重画小汽车做准备
            MOV     DI, POINTER
            MOV     DH, LINE
            CMP     DL, 80
```

```
                JAE     ENDL
                INC     COLUMN                  ；INC 向左走, DEC 向右走
                MOV     DL, COLUMN
        ENDL:   RET
ERASECAR        ENDP
；画树子程序
DRAWTREE        PROC
                PUSHR   AX, BX, CX, DX
                MOV     DX, 1608H               ；初始行列号
                MOV     CX, 26                  ；将小树树木送 CX
                MOV     AL, TREE                ；字符送 AL
                MOV     BL, TREE+1              ；字符属性送 BL
        TRE:    MOV     AH, 2                   ；置光标
                INT     10H
                PUSH    CX
                SETIOM  AH, 9, MOV, CX, 1, 10H  ；写字符
                ADD     DL, 2
                POP     CX
                LOOP    TRE
                POPR    AX, BX, CX, DX
                RET
DRAWTREE        ENDP
；由于道路两旁的树木有不同的初始行列号，故再次定义与上面程序功能一样的子程序
DRAWTRE1        PROC
                PUSHR   AX, BX, CX, DX
                MOV     DX, 1011H
                MOV     CX, 26
                MOV     AL, TREE
                MOV     BL, TREE+1
```

```
TRE1:   MOV     AH, 2
        INT     10H
        PUSH    CX
        SETIOM  AH, 9, MOV, CX, 1, 10H
        ADD     DL, 2
        POP     CX
        LOOP    TRE1
        POPR    AX, BX, CX, DX
        RET
DRAWTRE1 ENDP
；画大树（花）子程序
HUA     PROC
        LEA     DI, FLOWER
        PUSHR   AX, BX, CX, DX
        XOR     CH, CH
        MOV     CL, [DI]                ；字符个数送 CX
        INC     DI                      ；指向第一个显示符号
NEXTH:  ADD     DH, [DI+2]
        ADD     DL, [DI+3]
        MOV     AH, 2
        INT     10H                     ；设置光标位置
        MOV     AL, [DI]                ；取字符
        MOV     BL, [DI+1]              ；取字符属性
        PUSH    CX                      ；保存计数值
        ；设置写彩色字符
        MOV     AH, 9
        MOV     CX, 1
        INT     10H
        POP     CX                      ；恢复计数值
        ADD     DI, 4                   ；指向下一个显示字符
```

```
        LOOP    NEXTH                   ；大树(花)没画完转 NEXTH
        POPR    AX, BX, CX, DX
        RET
HUA     ENDP
；音乐子程序
MUSIC   PROC
        PUSHR   AX, BX, CX, DX
        MOV     SI, OFFSET FREQ
        MOV     DI, OFFSET TIME
  L1:   MOV     CX, [SI]                ；发声频率送 CX
        CMP     CX, 1
        JE      EXIT
        MOV     BX, [DI]
        PUSH    DX
        MOV     AL, 0B6H                ；设置定时器的控制寄存器，也就
                                        ；是将 0B6H 送到 43H 端口
        OUT     43H, AL                 ；发命令给寄存器
        MOV     DX, 08H
        MOV     AX, 3208H
        DIV     CX
        OUT     42H, AL                 ；初始值的低位字节和高位字节送
                                        ；到 42H 端口
        MOV     AL, AH
        OUT     42H, AL                 ；输出频率低字节
        IN      AL, 61H                 ；通过将 61H 端口的位 0 和位 1 设
                                        ；置为 1 将扬声器打开
        MOV     AH, AL
        OR      AL, 3
        OUT     61H, AL                 ；驱动扬声器
  L2:   PUSH    DX
```

```
        PUSH    AX
        MOV     DX, 8H
        MOV     AX, 0F05H
S1:     SUB     AX, 1
        SBB     DX, 0
        JNZ     S1
        POP     AX
        POP     DX
        DEC     BX
        JNZ     L2
        MOV     AL, AH
        OUT     61H, AL         ；将端口 61H 的位 0 和位 1 清零，
                                ；关闭扬声器
        POP     DX
        ADD     SI, 2
        ADD     DI, 2
        JMP     L1
EXIT:   POPR    AX, BX, CX, DX
        RET
MUSIC   ENDP
；延时子程序：DELAY
DELAY   PROC
        PUSHR   AX, BX, CX, DX
        MOV     DX, 7000        ；设置外循环计数值
GO:     MOV     CX, 20000       ；设置内循环计数值
REPEAT: LOOP    REPEAT          ；内循环是否结束
        DEC     DX
        JNE     GO              ；外循环是否结束
        POPR    AX, BX, CX, DX
        RET
```

```
DELAY       ENDP
；由于延时时间不一样，故再定义与上面程序功能一样的子程序：DELAY2
DELAY2      PROC
            PUSHR   AX, BX, CX, DX
            MOV     DX, 50000
      GO2:  MOV     CX, 50000
REPEAT2:    LOOP    REPEAT2
            DEC     DX
            JNE     GO2
            POPR    AX, BX, CX, DX
            RET
DELAY2      ENDP
；清屏幕子程序：CLEAR
CLEAR       PROC
            PUSHR   AX, BX, CX, DX
            MOV     BH, 7
            MOV     DX, 184FH
            MOV     AX, 0600H               ；用当前也上滚功能调用清屏幕
            MOV     CX, 0
            INT     10H
            POPR    AX, BX, CX, DX
            RET
CLEAR       ENDP
CODES       ENDS
            END     START
```

第 4 篇　模拟试卷

试卷一

一、单项选择题（本大题共 15 小题，每小题 2 分，共 30 分）

1．与 MOV　BX, OFFSET　VAR 指令完全等效的指令是（　　）。

（A）MOV　BX, VAR　　（B）LDS　　BX, VAR

（C）LES　　BX, VAR　　（D）LEA　　BX, VAR

2.　DEBUG 中的执行一段子程序的命令是（　　）。

（A）D　　（B）T　　（C）P　　（D）U

3．不能作为地址指针的寄存器是（　　）。

（A）CX　　（B）　BX　　（C）DI　　（D）SI

4．已知, （AX）=0F234H，执行 OR　AX, 963FH 后，则 SF，CF 的值为（　　）。

（A）SF=1，CF=1　　（B）SF=1，CF=0

（C）SF=0，CF=1　　（D）SF=0，CF=0

5．AL=11011011B ，CL=2，执行 SAR　AL, CL 后，AL 寄存器的内容是（　　）。

（A）00110110　　（B）01101100

（C）01101111　　（D）11110110

6．下面两个传送指令语句中源操作数寻址方式是（　　）。

VAR1　DW　32H

VAR2=32H

.....

MOV　　AX,　VAR1

MOV　　BX,　VAR2

（A）都是立即数寻址　　（B）立即数寻址，直接寻址

（C）直接寻址，立即数寻址　　（D）都是直接寻址

7．变量具有（　　）。

（A）偏移属性　　（B）　符号属性

（C）段属性　　（D）段属性和偏移属性

8．寄存器间接寻址方式中，操作数在（　　）中。

（A）通用寄存器　　（B）堆栈

（C）　内存单元　　（D）段寄存器

9．下列对 END 语句的叙述正确的是（　　）。

（A）END 是一可执行语句

（B）END 表示程序执行到此结束

（C）END 表示源程序到此结束

（D）END 在汇编后产生机器码

10．下列叙述不正确的是（　　）。

（A）存储单元与逻辑地址之间的关系是一一对应的

（B）存储单元与物理地址之间是一一对应的

（C）一个逻辑地址对应一个存储单元

（D）一个存储单元可以用不同的段地址和偏移地址表示

11．下列指令合法的是（　　）。

（A）OUT　AX, 30H　　（B）OUT　　DX, AL

（C）OUT　AL, DX　　（D）OUT　30H, AH

12．当一个带符号数大于 0FBH 时程序转移，选择的条件转移指令是（　　）。

（A）JNLE　　（B）JLE

（C）JNL　　（D）JA

13．用于显示单个字符的 DOS 系统功能调用，应设置 AH 寄存器的内容为（　　）。

（A）1H　　（B）2H

（C）9 H　　（D）0AH

14．完成与指令 XCHG　AX, BX 相同功能的指令或指令序列是（　　）。

（A）MOV　AX, BX　　（B）MOV　BX, AX

（C）PUSH　AX　　（D）MOV　CX, AX

POP　　BX　　MOV　　AX, BX

MOV　　BX, CX

15．下面的 CMP 指令中，语法正确的指令是（　　）.

（A）CMP　　BUF1, BUF2　　（B）CMP　　AL, 0FH

（C）CMP　BUF1, 0FH　　（D）CMP　AL, BX

二、阅读程序题（本大题共 3 小题，每小题 6 分，共 18 分）

1. 阅读程序

（1）写出程序的功能；（2）程序执行后 DAT 和 DAT+1 单元的内容是什么？

```
DATA    SEGMENT
DATDB   98H, 27H
DATA    ENDS
CODE    SEGMENT
        ASSUME    CS: CODE, DS: DATA
START:  MOV       AX, DATA
        MOV       DS, AX
        MOV       AL, DAT
        CMP       AL, DAT+1
        JNG       DONE
        XCHG      AL, DAT+1
        MOV       DAT, AL
DONE:   MOV       AH, 4CH
        INT       21H
CODE    ENDS
        END       START
```

2. 对于下面的数据定义，各条 MOV 指令单独执行后，请填写有关寄存器的内容。

```
DATA    SEGMENT
TABLE1  DW    30H
TABLE2  DW    20 DUP (?)
TABLE3  DB    'ABCD'
DATA    ENDS
.       MOV   BX, TYPE     TABLE2      （BX=_____）
        MOV   CX, LENGTH   TABLE2      （CX=_____）
        MOV   DX, OFFSET   TBLE3       （DX=_____）
```

3．阅读下列程序，写出指令段实现的功能。

```
MOV     AL, X
ADD     AL, Y
MOV     CL, 3
SAL     AL, CL
MOV     BL, X
SUB     BL, Y
SAL     BL, 1
SUB     AL, BL
MOV     CL, 4
SAR     AL, CL
MOV     Z, AL
```

三、阅读填空题（本大题共 2 小题，共有 6 个空，每个空格 2 分，共 12 分）

1．从 DAT 开始的单元中存放着多个有符号的字节数。下面程序实现对其处理。如果数据大于等于 0，则屏蔽高 4 位；如果小于 0 则将其求补，处理后的数据仍保留在原单元中。在下面每一空白处填上一条适当的指令。

```
DATAS   SEGMENT
DAT     DB   77H, 88H, 0AAH, 0BBH, 13H
COUNT   EQU   $-DAT
DATAS   ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATAS
START:  MOV     AX, DATAS
        MOV     DS, AX
        LEA     SI,   DAT
        MOV     CX, COUNT
LL1:    MOV     AL, [SI]
        CMP     AL, 0
        JGE     LL2
        ____________
```

```
        JMP     PP

LL2:    _____________
PP:     INC     SI
        LOOP    LL1
        MOV     AH,4CH
        INT     21H
SUB1    PROC
        NEG     AL
        MOV     [SI], AL
        RET
SUB1    ENDP
SUB2    PROC
        _____________
        _____________
        RET
SUB2    ENDP
CODE    ENDS
        END     START
```

2．下面程序段是完成用单个字符输入的 1 号功能调用，从键盘输入 10 个字符。然后再用单个字符显示输出的 2 号功能调用，并以相反顺序显示输出这 10 个字符。试在空白处填上适当的指令（每个空白处只能填一条指令）。

```
        MOV     CX, 10
LOP1:   MOV     AH, 01H
        INT     21H
        _____________
        LOOP    LOP1
        MOV     CX, 10
LOP2:   _____________
        MOV     AH, 02H
        INT     21H
```

LOOP　LOP2

四、程序设计题（本大题共 3 小题，其中第 1 小题 12 分，第 2 小题 13 分，第 3 小题 15 分，共 40 分）

1. X,Y, W 为字节类型的数，求 Z=（(W-X）*10+5）/（X+Y)），其中 X，Y，W 为字节类型的数，数值分别为：38H，28H，48H 。

2. 利用主程序调用子程序，比较 BUF1 和 BUF2 缓冲区中不相等的字符，并将不相等的字符显示出来。

3. 在首地址为 TAB 的数组中按递增次序存放着 100 个数，试编程将出现次数最多的数以及出现的次数分别存放于变量 NUM 和 COUNT 中，并以：COUNT=? 的形式输出 COUNT 的值（设最多重复次数<10） 。

试卷二

一、单项选择题（本大题共 20 小题，每小题 2 分，共 40 分）

1. 下列四个寄存器中，可作为 16 位寄存器的是（　　）。

（A）AL　　（B）BL　　（C）BP　　（D）BH

2. 逻辑移位指令 SHL，用于（　）。

（A）带符号数倍增　　（B）带符号数倍减

（C）无符号数倍增　　（D）无符号数倍减

3. 在程序的运行过程中，确定下一条指令的物理地址的计算表达式是（　　）。

（A）CS*16+IP　　（B） DX*16+DI

（C）SS*16+SP　　（D）ES*16+SI

4. 下面表示宏定义结束的命令是（　　）。

（A）ENDP　　（B）ENDS　　（C）ENDM　　（D）END

5. 用于输入单个字符的 DOS 系统功能调用，应设置 AH 寄存器的值为（　　）。

（A）1H　　（B）2H　　（C）9 H　　（D）0AH

6. 下列指令中有语法错误的是（　）。

（A）MOV　100H，BX　　（B）MOV　[100H], 25

（C）MOV　[100H], BX　　（D）MOV　[BX], 100H

7. 两个无符号数比较，如表示大于时转到 NEXT 处，应选用的指令是（　　）。

（A）JNBE　NEXT　　（B）JNLE　NEXT

（C）JBE　NEXT　　（D）JLE　NEXT

8. 寄存器直接寻址方式中，操作数在（　）中。

（A）通用寄存器　　（B）堆栈

（C）内存单元　　（D）段寄存器

9. 下列寄存器组中在段内寻址时可以提供偏移地址的寄存器组是（　　）。

（A）AX, BX, CX, DX　　（B）BX, BP, SI, DI

（C）SP, IP, BP, DX　　（D）CS, DS, ES, SS

10. 下面指令序列执行后完成的运算，正确的算术表达式是（　　）。

MOV　DL, X

SAR　　DL, 1

MOV　　Y, DL

（A）Y=X*2　　（B）X=Y*2　　（C）Y=X/2　　（D）X=Y/2

11. 在标志寄存器中，用以记录当前运算结果是否为 0 的标志位是（　　）。

（A）CF　　（B）ZF　　（C） OF　　（D）SF

12. 当前指令的地址存放在（　　）中

（A）DS: BP　　（B）SS: SP　　（C）CS: PC　　（D）CS: IP

13. 下列指令执行后，可能改变 BL 寄存器内容的指令是（　　）。

（A）TEST　BL, 0FFH　　（B）OR　　BL, 00H

（C）CMP　BL, 00H　　（D）XOR　BL, BL

14. 与 MOV　BX, OFFSET　VAR 指令完全等效的指令是（　　）。

（A）MOV　BX, VAR　　（B）LDS　　BX, VAR

（C）LES　　BX, VAR　　（D）LEA　　BX, VAR

15. 汇编语言源程序中，每个语句由四项组成，如语句要完成一定的功能，不可省略的项是（　　）。

（A）名字项　　（B）操作项

（C）操作数项　　（D）注释项

16. 汇编语言源程序的扩展名是（　　）。

（A）.ASM　　（B）.OBJ　　（C）.EXE　　（D）.COM

17. 能够将 BL 的低 4 位清 0 的指令是（　　）。

（A）AND　BL , 0F0H　　（B）AND　BL , 00H

（C）OR　　BL , 0F0H　　（D）OR　　BL , 00H

18. 已知，（AX）=9023H，执行 OR　AX, 001BH 后，则 SF，CF 的值为（　　）。

（A）SF=1，CF=1　　（B）SF=1，CF=0

（C）SF=0，CF=1　　（D）SF=0，CF=0

19. 设 0P1、0P2 是变量，下面哪条指令是正确的（　　）。

（A）CMP　BX, OP2　　（B）CMP　[AX], OP2

（C）CMP　OP1, OP2　　（D）CMP　OP2, 0FFH

20. 用 DEBUG 调试程序时，单步执行命令是（　　）。

（A）G　　（B）U　　（C）D　　（D） T

二、阅读程序题（本大题共 3 小题，其第 1、2 小题均为 6 分，第 3 小题为 6 分，共 20 分）

1．执行下面程序段后，AX、CX 的值为多少？

```
CODE    SEGMENT
        ASSUME      CS: CODE
START:  MOV     CH, 0
        MOV     CL, 1
        MOV     BL, 2
        MOV     AH, 0
AGAIN:  MOV     AL, BL
        INC     BL
        MUL     BL
        ADD     CX, AX
        CMP     AX, 0015H
        JB      AGAIN
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

2．阅读程序，写出指令段实现的数学公式。

```
MOV     BL, X
ADD     BL, Y
MOV     AL, W
SUB     AL, X
IMUL    TEN
ADD     AX, 5
IDIV    BL
MOV     Z, AX
MOV     AL, 0
```

3. 写出程序的功能和结果。

```
DSEG    SEGMENT
```

```
NUM1    DB  1, 2, 3 ,4, 5, 6, 7, 8
NUM2    DB  1, 2, 3, 4, 5, 6, 7, 8
RESULT DW ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME      CS: CSEG, DS: DSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        MOV     CX, 8
        MOV     BX, 0
        MOV     SI, OFFSET NUM1
        MOV     DI, OFFSET NUM2
AGAIN:  MOV     AL, [SI]
        MUL     BYTE PTR[DI]
        ADD     BX, AX
        INC     DI
        INC     SI
        LOOP    AGAIN
        MOV     RESULT, BX
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
CSEG    ENDS
        END     START
```

三、程序设计题（本大题共 3 小题，其中第 1 小题 12 分，第 2 小题 13 分，第 3 小题 15 分，共 40 分）

1．编程显示以下图案。

```
*********
*******
*****
```

***

*

2．首地址为 BUF 的内存单元中存有 10 个字节数，统计其中单元内容为 0 的单元个数，并将统计的结果显示出来。

3．编写子程序搜索指定字符缓冲区中是否有小写字母 N，如果有则用 Y 替代。江调用子程序将 BUF1，BUF2 中的 N 全部用 Y 替代。

试卷三

一、单项选择题（本大题共 20 小题，每小题 1 分，共 20 分）

1．下面四个寄存器中，不能作为间接寻址的寄存器是（　　）。

（A）BX　　（B）DX　　（C）BP　　（D）DI

2．用来表示堆栈指针的寄存器是（　　）。

（A）IP　　（B）BP　　（C）SP　　（D）SS

3．完成 BX 清零，并使标志位 CF 清零，下面错误的指令是（　　）。

（A）SUB　BX, BX　　（B）XOR　BX, BX

（C）MOV　BX, 00H　　（D）AMD　BX, 00H

4．下面数据传送指令中，正确的指令是（　　）。

（A）MOV　BUF1, BUF2　　（B）MOV　CS, AX

（C）MOV　CL, 1000H　　（D）MOV　DX, WORD PTR [BX+DI]

5．下面指令中，源操作数的寻址方式为直接寻址的指令是（　　）。

（A）ADD　AX, WORD PTR [BX+SI]　　（B）ADD　AX, B

（C）INC　CX, 1000H　　（D）MOV　BX, 7FFFH

6．下面表示段定义结束的命令是（　　）。

（A）ENDP　　（B）ENDS　　（C）ENDM　　（D）END

7．设 AL,BL 中都是有符号数，当 AL≤BL 时转至 NEXT 处，在 CMP　AL, BL 指令后应选用正确的条件转移指令是（　　）。

（A）JBE　　（B）JNC　　（C）JNA　　（D）JNLE

8．指令 LOOPZ 的循环执行条件是（　　）。

（A）CX≠0 并且 ZH=0　　（B）CX≠0 或 ZH=0

（C）CX≠0 并且 ZH=1　　（D）CX≠0 或 ZH=1

9．执行下列指令后，SP 寄存器的值是（　　）。

MOV　SP, 1000H

PUSH　AX

（A）0FFEH　　（B）0FFFH　　（C）1001H　　（D）1002H

10．完成有符号数 BX 的内容除以 2 的正确指令是（　　）。

（A）SHR　BX, 1　　（B）SAR　BX, 1

（C）ROR　BX, 1　　（D）RCR　BX, 1

11．下面指令中影响状态标志位 CF 的指令是（　　）。

（A）INC　AX　　（B）DEC　AX

（C）NOT　AX　　（D）DEC AX

12．判断当 CX=0 时，转 ZERO 执行的错误指令是（　　）。

（A）CMP CX, 0
　　JZ　ZERO

（B）MOV　AX, CX
　　JZ　ZERO

（C）OR　CX, CX
　　JZ　ZERO

（D）JCXZ　ZERO

13．执行 1 号 DOS 系统功能调用后，从键盘输入的字符应存放在（　　）。

（A）AL 中　　（B）BL 中　　（C）CL 中　　（D）DL 中

14．设数据定义如下：

```
ARRAY    DW   1, 2, 3
```

执行指令 ADD　SI, TYPE　ARRAY 相当于完成下面同等功能的指令是（　　）。

（A）ADD　SI, 1　　（B）ADD　SI, 2

（C）ADD　SI, 3　　（D）ADD　SI, 0

15．设数据定义如下：

```
BUF    DW   100 DUP (?)
```

执行指令 MOV　CX, LENGTH　BUF 相当于完成下面同等功能的指令是（　　）。

（A）MOV　CX, 00C8H　　（B）MOV　CX, 0200H

（C）MOV　CX, 0064H　　（D）MOV　CX, 0100H

16．完成对字单元 BUF 的内容加 1 运算，下面错误的指令是（　　）。

（A）MOV　CX, BUF
　　INC　BX
　　MOV　BUF, BX

（B）MOV　BX, OFFSET BUF
　　INC　WORD　PTR [BX]

（C）MOV　BX, BUF
　　INC　WORD　PTR [BX]

（D）MOV　BX, 0
　　INC　WORD　PTR BUF [BX]

17．完成对寄存器 BX 内容的求补运算，下面错误的指令是（　　）。

（A）NEG　BX

（B）NOT　BX
　　INC　BX

（C）XOR　BX, 0FFFFH
　　INC　BX

（D）MOV　AX, 0
　　SUB　AX, BX

18．判断当 AX 的内容为负数时，转 MINUS 执行，下面错误的指令是（　　）。

（A）NOT　AX
　　JNS　MINUS

（B）TEST　AX, 8000H
　　JNS　MINUS

（C）SHL　AX, 1
　　JC　MINUS

（D）OR　AX, AX
　　JNS　MINUS

19．下面指令执行后，正确的结果是（　　）。

```
MOV　AL, 0FEH
ADD　AL, AL
ADC　AL, AL
```

（A）AL=0F8H　CF=1

（B）AL=0F8H　CF=0

（C）AL=0F9H　CF=0

（D）AL=0F9H　CF=1

20．现有数据存储如下图所示：设 AL=01H，BX=0100H，DX=3000H，执行换码指令 XLAT 后正确的结果是（　　）。

30100H	40H
30101H	79H
30102H	24H
30103H	30H

（A）AL=00H　（B）AL=40H　（C）AL=01H　（D）AL=79H

二、填空题（本大题共 6 小题，每空 1 分，共 14 分）

1. 8086CPU 允许的最大存储空间为_____，其地址编号从_____到_____H。

2．汇编语言是一种面向_____的语言。完成将汇编语言源程序翻译成机器语言目的程序的翻译程序称_____。

3．8086 的存储器是分段的，因此存储单元的物理地址是由_____和组合_____而成的。

4．标号和变量（名字）均有三种属性，它们是_____、_____和_____属性

5．判断无符号运算是否溢出，应根据_____标志位；而判断有符号数运算是否溢出，应根据_____标志位。

6．在 16 位存储单元中能存储的最大无符号数为 65535，能表示的有符号数的范围是_____ H 到_____ H。

三、简答题（本大题共 3 小题，每小题 3 分，共 9 分）

1．数据存储示意如下图所示，请用 DUP 语句写出合适的数据定义伪指令：

BUFFER

1
2
3
2
3
4
1
2
3
2
3
4

2．选用合适的指令，分别完成下列操作：

A）将字变量 VARW 的偏移地址送 BX 寄存器；

B）将字变量 VARW 的内容送 CX 寄存器；

C）将字变量 VARW 的类型值送 DX 寄存器。

3．设有数据定义伪指令如下：

```
ORG     2000H
ARRAY  DW   0100H, 0200H, $+2, 0300H, $+2
```

请画出存储单元数据的存储形式。

四、程序分析题（本大题共 5 小题，每小题 6 分，共 30 分）

1．
```
MOV     AX, 00FFH
MOV     BX, 0FFFFH
XOR     AX, BX
NEG     AX
```
问：该程序段执行后 AX=_____，CF=_____。

2．
```
MOV     AX, BX
NEG     AX
ADD     AX, BX
```
问：该程序执行后 AX=_____，CF=_____。

3．
```
BUF     DW      0000H
        …
        LEA     BX, BUF
        STC
        RCR     WORD  PTR     [BX], 1
        MOV     CL, 3
        SAR     WORD  PTR     [BX], CL
```
问：该程序执行后，存储单元 BUF 的内容为_____。

4．
```
BLOCK     DB      20H, 1FH, 08H, 81H, 0FFH…
RESULT    DB      ?
          …
START:    LEA     SI, BLOCK
          MOV     CX, [SI]
          INC     SI
          MOV     AL, [SI]
LOP1:     CMP     AL, [SI+1]
          JNG     NEXT
          MOV     AL, [SI+1]
NEXT:     INC     SI
          LOOP    LOP1
          MOV     RESULT,AL
```

```
        HLT
```

问：（1）该程序所完成的功能是_____；

（2）该程序循环次数是：_____。

5．
```
CODE    SEGMENT
        ASSUME  CS: CODE
START:  MOV     CX, 1
        MOV     BL, 2
AGAIN:  MOV     AL, BL
        INC     BL
        INC     BL
        ADD     CX, AX
        CMP     AX, 002AH
        JB      AGAIN
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

问：（1）该程序所完成的功能可用算术表达式表示为_____；

（2）该程序完成后 CX=_____。

五、程序填空题（本大题共 2 小题，每小题 6 分，共 12 分）

（下列各小题中，每空只能填一条指令）

1．下面程序完成十进制数 3298+4651 的运算，并将结果存入 SUM 单元开始的 2 个字节单元中，请将程序补充完整。

```
DA1     DB  98H, 32H
DA2     DB  51H, 46H
SUM     DB  2   DUP (?)
        …
        MOV     SI, OFFSET  DAT
        LEA     DI, DAT
        _____
```

```
        ADD     AL, [DI]
        _____________________
        MOV     SUM, AL
        MOV     AL,  [SI+1]
        _____________________
        DAA
        MOV     SUM+1, AL
        HLT
```

2．下面程序利用 DOS 系统功能调用，完成将键盘输入的小写字母转换成大写字母后输出显示，直到输入'$'字符结束。请将程序补充完整。

```
CODE        SEGMENT
            ASSUME      CS: CODE
DISPLAY     PROC    NEAR
BEGIN       MOV     AH, 01H
            INT     21H
            JZ      STOP
            CMP     AL, 'a'
            JB      STOP
            CMP     AL, 'z'
            JA      STOP
            _____________________
            _____________________
            MOV     AH, 02H
            INT     21H
            JMP     BEGIN
STOP:       RET
DISPLAY     ENDP
CODE        ENDS
```

六、程序设计题（本大题共 2 小题，其中第 1 小题 5 分，第人小题 10 分，共 15 分）

1．设在 DAT1，DAT2 字单元中存放一双字长有符号数，编一程序段，完成求出该双字长

数的绝对值后送 ABS1 和 ABS2 字存储单元。（本小题 5 分）

2．在附加数据段中有一首址为 ADDR 的没有排序的字数组，数据的第一个字为数组长度，第二个字开始存放数组各元素。在 KEY 单元存放一要删除的字数据。要求编一完整程序，在数组中查找该数，如果找到该数，将其从数组中删除，并修改数组长度。（本小题 10 分）

试卷一参考答案

一、填空题

1 D　2 C　3 A　4 B　5 D　6 C　7 D　8 C　9C　10 A　11 B　12 A　13 B　14 B　15 B

二、阅读题

1.（1）比较 DAT 与 DAT+1 单元的内容，小的数放在 DAT 单元

（2）27H，98H

2. BX=2　CX=20　DX=42

3. Z＝（（X+Y）*8-（X-Y）*2）/16

三、程序填空题

1. CALL　SUB2　CALL　SUB1　AND　AL, 0FH　MOV　[SI] , AL

2. PUSH　AX　POP　DX

四、程序设计题（参考答案，也可以用其它方法）

```
1.    DSEG   SEGMENT
          X   DB    38H
          Y   DB    20H
          W   DB    30H
          Z   DW    ?
       TEN    DB    10
      DSEG    ENDS
      CSEG    SEGMENT    PARA    PUBLIC 'CODE'
              ASSUME     CS: CSEG, DS: DSEG
      START:  MOV    AX, DSEG
              MOV    DS, AX
              MOV    BL, X
              ADD    BL, Y                    ；先求分母，存于 BL
              MOV    AL, W
              SUB    AL, X
              IMUL   TEN
```

```
        ADD     AX, 5
        IDIV    BL
        MOV     Z, AX
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
CSEG    ENDS
        END     START
```

2.
```
DATA    SEGMENT
 BUF1   DB 'HUIFEHJFKHKFJ'
 LEN1   =$-BUF1
 BUF2   DB 'HUIFEHJFIOIHJ'
 BUF    DB LEN1 DUP (?)                ；不相同的字符存在 BUF 开始的缓冲区
DATA    ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        LEA     DI, BUF1
        LEA     SI, BUF2
        MOV     CX, LEN1
        LEA     BX, BUF
        CALL    KLL
        MOV     BYTE PTR[BX], '$'       ；在要显示的字符串末尾加'$'
                                        ：显示字符串 BUF
        LEA     DX, BUF
        CALL    PRINT
        MOV     AH, 4CH
        INT     21H
```

；比较 2 个字符串，将不相同的字符送到 BX 开始的存储空间，入口参数为 2 个字符

串的偏移地址 SI，DI，以及字符串的长度 CX

```
KLL     PROC
AL1:    MOV     AL, [SI]
        CMP     [DI], AL
        JE      AL2
        MOV     [BX], AL
        INC     BX
AL2:    INC     SI
        INC     DI
        LOOP    AL1
        RET
KLL     ENDP
PRINT   PROC    NEAR                ；显示字符串子程序
        MOV     AH, 9
        INT     21H
        RET
PRINT   ENDP
CODE    ENDS
        END     START
```

3. 在首地址为 TAB 的数组中按递增次序存放着 100 个数，试编程将出现次数最多的数以及出现的次数分别存放于变量 NUM 和 COUNT 中，并以：COUNT=? 的形式输出 COUNT 的值（设最多重复次数<10）。

```
DATA    SEGMENT
BUF     DB 1, 2, 14H, 14H, 14H, 14H, 14H, 15H, 15H, 15H, 17H, 17H, 59H, ...
LEN     = $-BUF
NUM     DB  ?
COUNT   DB  0
MM      DB 'COUNT=', '$'
DATA    ENDS
CODE    SEGMENT
```

```
        ASSUME CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     SI, 0
        MOV     CL, LEN
        MOV     BL, 0
AGAIN:  MOV     AL, BUF[SI]
        CMP     AL, BUF[SI+1]
        JNE     KL1
        INC     BL              ；BL 中存放的是当前重复数的个数
        MOV     BH, AL          ；BH 中存放的是当前重复的那个数
        JMP     KL2
KL1:    CMP     COUNT, BL       ；重复次数最多的放在 COUNT 中
        JAE     L2
        MOV     COUNT, BL
        MOV     NUM, BH         ；保留重复次数最多的那个数
        MOV     BL, 0           ；要将 BL 置 0, 用于存放下一个重复数
                                ；的个数
KL2:    INC     SI
        LOOP    AGAIN
        INC     COUNT           ；COUNT 中是重复的次数，出现次数
                                ；应该是重复次数加 1
        LEA     DX, MM
        MOV     AH, 9
        INT     21H
        MOV     DL, COUNT
        ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     AH, 4CH
```

```
        INT     21H
CODE    ENDS
        END     START
```

试卷二参考答案

一、选择题

1 C　2 C　3 A　4 C　5 A　6 A　7 A　8 C　9 B　10 C　11 B　12 D　13 D　14 D　15 B

16A　17 A　18 B　19 A　20 D

二、阅读题

1.　AX　　CX　　（也可以用十进制数表示）

6　　7

C　　13H

14H　　27H

1EH　　45H

2.　（（X+Y）*8-（X-Y）*2）/16

3.　Y＝∑AI*BI，I=1—8（用中文表示也可以）

Y=204

三、编程题

1.

```
CSEG    SEGMENT
        ASSUME    CS: CSEG
START:  MOV     AX, DSEG
        MOV     DS, AX
        MOV     BX, 9
LP:     MOV     CX, BX
LOP:    MOV     DL, '*'
        MOV     AH, 2
        INT     21H
        LOOP    LOP
        MOV     AH, 02H
        MOV     DL, 0AH
```

```
        INT     21H
        MOV     DL, 0DH
        INT     21H
        DEC     BX
        DEC     BX
        CMP     BX, 1
        JGE     LP
        MOV     AL, 0
        MOV     AH, 4CH
        INT     21H
CSEG    ENDS
        END     START
```

2.
```
DATA    SEGMENT
BUF     DB 0, -1, 2, 4, 3, -4, 5, 6, -7, 0
RESULT DB   0                           ；存放 0 的个数
DATA    ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AL, 0                   ；计数 0 的个数
        LEA     SI, BUF
        MOV     CX, 10
LOP:    CMP     BYTE PTR[SI], 0
        JNE     AL3
        INC     AL                      ；0 的个数
AL3:    INC     SI
        LOOP    LOP
        MOV     DL, AL
        ADD     DL, 30H
```

```
        MOV     AH, 02H
        INT     21H
        MOV     RESULT, AL
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

3.
```
DATA    SEGMENT
BUF1    DB 'I AM A STUDENT!', '$'
LEN1    =$-BUF1
BUF2    DB 'EWFHEFNFN', '$'
LEN2    =$-BUF2
M       DB 13, 10, '$'
DATA    ENDS
CODE    SEGMENT
        ASSUME      CS: CODE, DS: DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        LEA     SI, BUF1
        MOV     CX, LEN1
        CALL    CMMP
        LEA     SI, BUF2
        MOV     CX, LEN2
        CALL    CMMP
        MOV     AH, 4CH
        INT     21H
```

；子程序的功能：搜索字符串中是否有 N 字符，如果有用 Y 替代，并显示字符串

；入口参数为字符串偏移地址 SI，字符串长度 CX

```
CMMP    PROC
        MOV    DX, SI
```

；将要显示的字符串的偏移地址送 DX，用于显示字符串。这句应在子程序的开始，因为下面语句会改变 SI 的值

```
        BL1: CMP    BYTE PTR[SI], 'N'
             JNZ    BL2
             MOV    AL, 'Y'
             MOV    [SI], AL
        BL2:INC     SI
             LOOP   B11
             MOV    AH, 09H                  ；显示字符串
             INT    21H
             LEA    DX, M
             INT    21H
             RET
    CMMP    ENDP
    CODE    ENDS
             END    START
```

试卷三参考答案

一、单项选择题

1. B　2. C　3. C　4. D　5. B　6. B　7. B　8. C　9. A　10. B　11. D　12. B　13. A　14. B

15. C　16. C　17. D　18. A　19. D　20. D

二、填空题

1. 1MB　00000H　FFFFFH　　2. 机器　汇编语言　　3. 段地址　偏移地址

4. 段属性 SEG　偏移地址属性 OFFSET　类型属性 TYPE

5. CF　OF　　6. 8000H　7FFFH

三、简答题

1. BUFFER　DB　2 DUP（1, 2DUP（2, 3）, 4）

2. A）MOV　BX, OFFSET VARW

B）MOV　CX, VARW

C）MOV　BX, TYPE VARW

3.

2000H	00H
2001H	01H
2002H	00H
2003H	02H
2004H	04H
2005H	20H
2006H	00H
2007H	03H
2008H	08H
2009H	20H

四、程序分析题

1. AX=0100H, CF=1

2. AX=0000H　　CF=0（当 BX=0 时）, CF=1（当 BX 不等于 0 时）

3. 该程序执行后，存储单元 BUF 的内容为 0F000H，即该程序段完成：-32768/8=-4096

4.（1）该程序完成的功能是从 32 个有符号数中找出最小数→RESULT 单元；

（2）该程序的循环次数是 1FH(或 31)

5.（1）该程序所完成的功能可用算术表达式表示为：

CX←1+2 x 3+3 x 4+4 x 5+5 x 6+6 x 7；

（2）程序运行后 CX=6FH（或 111）

五、程序填空题

1. 对该程序所要求补充的指令分别是：MOV AL, [SI]

DAA

ADC AL, [DI+1]

2. CMP AL, '$'

SUB AL, 20H

MOV DL, AL

六、程序设计题

```
1.  START:  MOV     AX, DAT1
            MOV     DX, DAT2
            OR      DX, DX
            JNS     NEXT
            NOT     AX
            NOT     DX
            ADD     AX, 1
            ADC     DX, 0
      NEXT: MOV     ABS1, AX
            MOV     ABS1, DX
            HLT
2.  DATA    SEGMENT
    ADDR    DW 1275H, 4652H, 0034H, 4CBAH, 1FF0H, …
    KEY     DW      0034H
    DATA    ENDS
    CODE    SEGMENT
            ASSUME      CS: CODE, DS:DTAT, ES:DATA
```

```
DELUL  PROC   FAR
       PUSH   DS
       XOR    AX, AX
       PUSH   AX
       MOV    AX, DATA
       MOV    DS, AX
       MOV    ES, AX
       LEA    DI, ADDR
       PUSH   DI
       MOV    AX, KEY
       MOV    CX, ES: [DI]
       ADD    DI, 2
       CLD
       REPNZ  SCASW
       JZ     DELET
       POP    DI
       JMP    EXIT
DELET: JCXZ   NEXT
DEL:   MOV    BX, ES: [DI]
       MOV    ES:[DI-2], BX
       ADD    DI, 2
       LOOP   DEL
NEXT:  POP    DI
       DEC    WORD  PTR     ES:[DI]
EXIT:  RET
DELUL  ENDP
CODE   ENDS
       END
```

参考文献

[1] 曹加恒等. 新一代汇编语言课程设计. 北京：高等教育出版社，2003

[2] 龚尚福等. 微型计算机汇编语言课程设计. 西安：西安电子科技大学出版社，2003

[3] 李珍香等. 汇编语言课程设计案例精编. 北京：中国水利水电出版社，2004

[4] 王爽. 汇编语言. 北京：清华大学出版社，2003

[5] 张维勇. 汇编语言程序设计自考应试指导. 南京：南京大学出版社，2000

[6] 罗万钧. 汇编语言程序设计教学辅导与上机实验指导. 西安：西安电子科技大学出版社，1998

[7] 李兆凤. 8088/8086 汇编语言程序设计. 北京：中央广播大学出版社，1993

[8] 杨路明. 汇编语言程序设计. 长沙：中南大学出版社，2005