

万水

课程设计丛书

汇编语言

课程设计 案例精编

李珍香 刘红梅 赵润林 编著



中国水利水电出版社
www.waterpub.com.cn

万水课程设计丛书

汇编语言课程设计案例精编

李珍香 刘红梅 赵润林 编著

中国水利水电出版社

内 容 提 要

本书共分 10 章, 列举了 27 个汇编语言在算术运算、非数值处理、字符图形设计、乐曲程序设计、动画程序设计、中断程序设计、系统功能调用、磁盘文件管理等方面的实用案例。本书所有程序已在 PC 系列微机上调试通过。所有案例的源程序代码和可执行程序都可从中国水利水电出版社网站下载, 网址为: www.waterpub.com.cn。

本书是一本实用的汇编语言课程设计教材, 可作为《汇编语言程序设计》的配套教材, 适合高等院校计算机专业或其他相关专业的学生使用, 同时对于使用汇编语言进行程序开发的技术人员也有一定的参考价值。

图书在版编目 (CIP) 数据

汇编语言课程设计案例精编/李珍香, 刘红梅, 赵润林编著. —北京: 中国水利水电出版社, 2004.3

(万水课程设计丛书)

ISBN 7-5084-2033-0

I. 汇… II. ①李…②刘…③赵… III. 汇编语言—程序设计—高等学校—教材 IV. TP313

中国版本图书馆 CIP 数据核字 (2004) 第 012751 号

书 名	汇编语言课程设计案例精编
作 者	李珍香 刘红梅 赵润林 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
经 售	
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 13 印张 294 千字
版 次	2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷
印 数	0001—5000 册
定 价	18.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

《汇编语言程序设计》作为一门课程，对于训练学生掌握程序设计技术、熟练上机操作及程序调试等技能有着重要的作用，但该课程实践性很强，只有通过实际编程和上机实践才能掌握程序设计技术，并使其达到较高水平。目前，市面上现有的侧重于实践技能的教材不能完全满足教学需求，为此，我们编写了这本汇编语言案例书，一方面作为《汇编语言程序设计》的配套教材为课程设计教学服务；另一方面，使用汇编语言进行程序开发的工程技术人员也可以根据本书内容进行上机实践，从而从中得到收益及提高。

为使本书成为一本有较高参考价值、培养学生汇编语言实用编程能力的教材，根据作者在多年的教学及工程实践中开发汇编语言程序的经验，本书以专题的方式，本着由浅入深的原则安排各章的内容。本书共 10 章，列举了 27 个汇编语言在算术运算、非数值处理、字符图形设计、乐曲程序设计、动画程序设计、中断程序设计、系统功能调用、磁盘文件管理等方面的实用案例。其中第 10 章是两个综合型的大程序，涵盖了《汇编语言程序设计》的所有知识，同时又为使用汇编语言程序设计者提供了编程技巧和思路。本书是一本实践性很强的教材。

本书所有程序都已在 PC 系列微机上调试通过。所有案例的源程序代码和可执行程序都可从中国水利水电出版社网站下载，网址为 www.waterpub.com.cn。

本书由李珍香总体规划和统稿，参加编写的还有刘红梅、赵润林同志。具体分工如下：案例 1、2、12、14、15、16、19 和 21 由刘红梅编写，案例 3、4、5、6、7、9、11、13、17、18、20、23、26 和 27 由李珍香编写，案例 8、10、22、24 和 25 由赵润林编写。在本书源程序的编写和调试过程中，曾得到多人的帮助；个别案例摘自于互联网，作者不详，在此一并对他们表示感谢。

编者在编写本书的过程中尽了很大的努力，但由于水平有限，错误和不妥之处在所难免，恳请广大读者批评指正。

编 者

2004 年 1 月

目 录

前言

第 1 章 算术运算程序设计	1
案例 1 判断闰年程序	1
案例 2 产生随机数并运算的程序	3
案例 3 计算平台长度程序	6
案例 4 加减乘除四则计算	9
第 2 章 非数值处理程序设计	17
案例 5 数字方阵程序	17
案例 6 数据加密程序	19
案例 7 代码转换程序	22
案例 8 键盘录入数据的转换与显示程序	27
第 3 章 发声程序设计	32
案例 9 音乐演奏程序	32
第 4 章 图形显示程序设计	38
案例 10 显示彩色矩形块的程序	38
案例 11 字符图形程序 (1) ——中国北京	39
案例 12 字符图形程序 (2) ——阳泉学院	42
案例 13 图形变换程序	48
第 5 章 动画程序设计	53
案例 14 汽车小鸟动画程序	53
案例 15 动画“心”程序	57
第 6 章 键盘输入程序设计	66
案例 16 打字练习	66
案例 17 打字计时练习	72
第 7 章 显示系统时间程序设计	79
案例 18 系统时间显示程序	79
案例 19 显示日期时间程序	81
案例 20 七段码电子表显示程序	84
第 8 章 文件管理程序设计	91
案例 21 简易文本编辑器	91
案例 22 磁盘文件的读写	101
案例 23 学生成绩管理程序	103

第 9 章 PC 机间的相互通信程序设计 114

 案例 24 用查询方式实现 PC 机间的相互通信 114

 案例 25 用中断方式实现 PC 机间的相互通信 118

第 10 章 综合程序设计 121

 案例 26 钟表显示 121

 案例 27 “霓红灯”的模拟显示 140

第 1 章 算术运算程序设计

算术运算是汇编语言程序设计中的重要组成部分。在实际应用中，不论是简单的还是复杂的程序，都或多或少地要用到一些算术运算，这些算术运算大多数是加、减、乘、除和乘方等。针对不同的运算对象，如多字节的数，不同码制的数，它们的运算方法和编程方法不同。即使对于同一种运算，在实现上也可以有多种方法。所以，用汇编语言编写算术运算程序有很多问题需要讨论和研究。

本章以定点数为主，列举了以下几个算术运算程序设计的例子。

案例 1 判断闰年程序

设计内容：判断某年是否为闰年。

要求：年份从键盘输入，通过运行计算后，输出相应信息。

设计思路：利用 DOS 21H 中断类型的 10 号键盘功能调用，将从键盘接收的年份数值型字符串按字节存入缓冲区。将字符串中的每个字符转换为对应的十进制数值（利用 0~9 这 10 个数的 ASCII 码值比其本身大 30H 的关系），通过判断某年是否为闰年的表达式，判断输出是否为闰年的信息。

本程序中判断某年是否为闰年的表达式为：

$((AX) \bmod 4 \text{ EQ } 0) \text{ AND } ((AX) \bmod 100 \text{ NE } 0) \text{ OR } ((AX) \bmod 400 \text{ EQ } 0)$

其中，(AX) 中存放的是从键盘输入的年份值。

本表达式的意思是：如果某一年能被 4 整除但不能被 100 整除或者能被 400 整除，则该年为闰年。

例如：从键盘输入“2003”，则从屏幕输出“This is not a leap year”信息。

本程序的流程图如图 1-1 所示。

操作说明：该程序须在 DOS 下运行。

源程序（Ifleap.asm）如下：

data	segment		db 200 dup(0)
infor	db 0dh,0ah,'Please input a year: \$'	stack	ends
y	db 0dh,0ah,'this is a leap year! \$'	code	segment
n	db 0dh,0ah,'This is not a leap year! \$'		assume ds:data,ss:stack,cs:code
w	dw 0	start:	mov ax,data
buf	db 8		mov ds,ax
	db ?		lea dx,infor ;在屏幕上显示
	db 8 dup(?)		;提示信息
data	ends		mov ah,9
stack	segment stack		int 21h

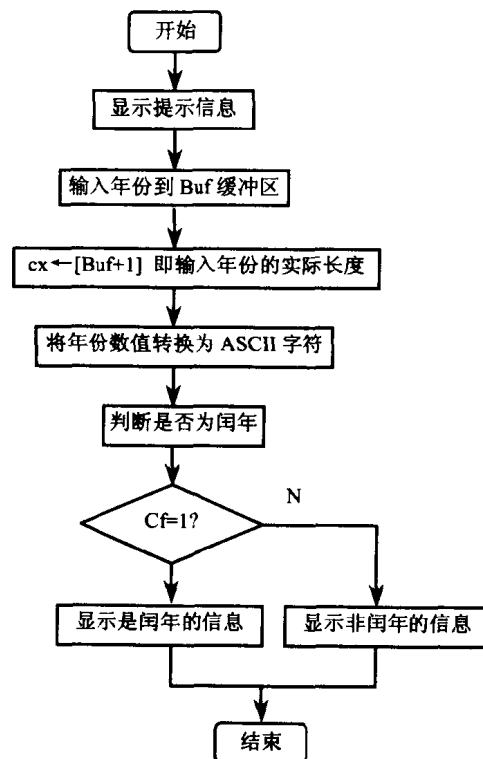


图 1-1 判断是否为闰年的程序结构图

<pre> ;字符串 lea dx,buf ;从键盘输入年份 mov ah,10 int 21h mov cl,[buf+1] lea di,buf+2 call datacate call ifyears jc al lea dx,n mov ah,9 int 21h jmp exit al: lea dx,y mov ah,9 int 21h exit: mov ah,4ch int 21h datacate proc near ;将数值转换为 ;ASCII 码字符子程序 push cx; dec cx lea si,buf+2 </pre>	<pre> tt1: inc si loop tt1 pop cx mov dh,30h mov bl,10 mov ax,1 l1: push ax sub byte ptr [si],dh mul byte ptr [si] add w,ax pop ax mul bl dec si loop l1 ret datacate endp ifyears proc near push bx push cx push dx mov ax,w mov cx,ax mov dx,0 </pre>
---	--

mov bx,4	cmp dx,0
div bx	jz lab2
cmp dx,0	lab1: cld
jnz lab1	jmp lab3
mov ax,cx	lab2: stc
mov bx,100	lab3: pop dx
div bx	pop cx
cmp dx,0	pop bx
jnz lab2	ret
mov ax,cx	ifyears endp
mov bx,400	code ends
div bx	end start

程序运行的结果如图 1-2 所示。

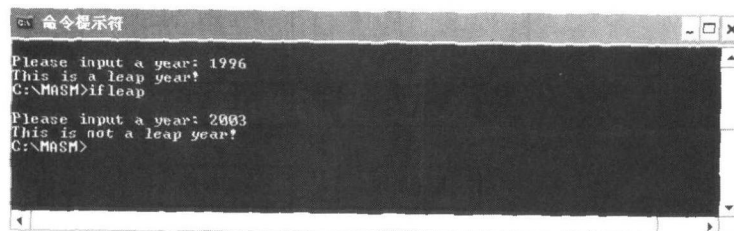


图 1-2 程序运行结果

案例 2 产生随机数并运算的程序

设计内容：编程实现产生两个随机数，并能对产生的随机数进行加法和减法运算。

设计思路：利用 BIOS 功能调用的时钟中断调用（1AH 中断调用），读取时钟计数器的值，利用随时变化的时间计数值产生随机数并进行加法或减法运算。然后将产生的随机数和运算结果转换为相应的十六进制数表示的 ASCII 码字符，并将其显示在屏幕上。

本程序流程图如图 1-3 所示。

操作说明：本程序须在 DOS 下运行。每运行一次可执行程序 random.exe，即可显示随机产生的两个十六进制数及它们的和与差。在减法运算中，如果被减数小于减数，则显示“Divide Overflow”的提示信息。

源程序（random.asm）如下：

data	segment	bnm dw 0	
	cc db '+'	data	ends
	ww db '-'	stack	segment
	gg db '='		db 200 dup(0)
	ii db 0dh,0ah,'\$'	stack	ends
	aa db 5 dup(?)	code	segment
	www dw 0		assume cs:code,ds:data,ss:stack
	ghh db '0',0dh,0ah,'\$'	start:	mov ax,data

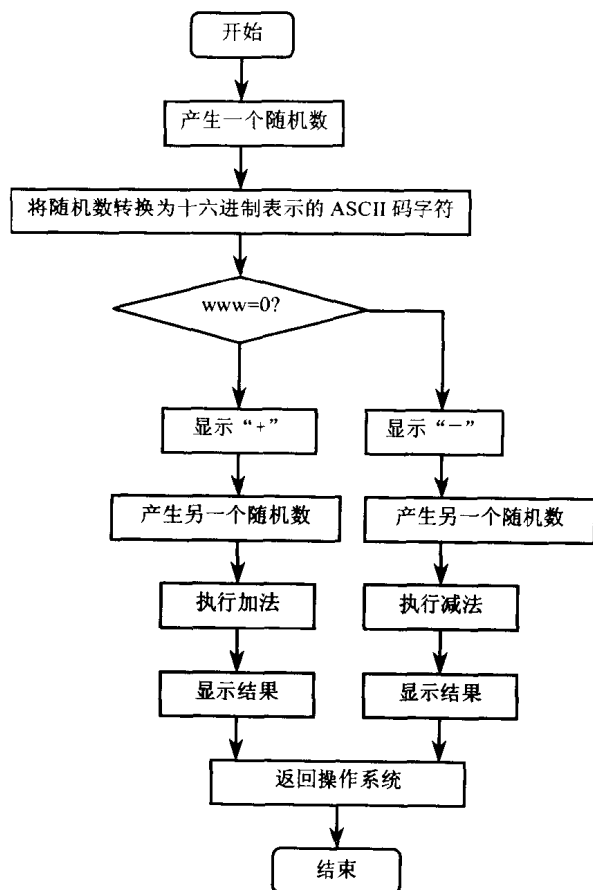


图 1-3 产生随机数程序流程图

<pre> mov ds,ax mov cx,20 call rand ;产生一个随机数 call mcat ;将随机数转换为 ;十六进制表示的 ASCII 码字符 mov ax,bx push ax cmp www,0 ;www=0? je a1 ;执行加法运算 cmp www,1 ;www=1? je a2 ;执行减法运算 a1: ;显示加号 mov dl,cc mov ah,2 int 21h call rand ;产生另一个随机数 add bx,0fh call mcat ;将随机数转换为 ;十六进制表示的 ASCII 码字符 mov dl,gg ;显示等号 mov ah,2 </pre>	<pre> int 21h pop ax add ax,bx cmp ax,0 jne yjw1 lea dx,ghh mov ah,9 int 21h jmp qq1 yjw1: jmp b1 a2: mov dl,ww ;显示减号 mov ah,2 int 21h mov cx,65535 mail: loop mail call randl call mcat mov dl,gg mov ah,2 int 21h </pre>
--	---

	pop ax		jmp s3
	sbb ax,bx	s1:	mov cl,4
	cmp ax,0		mov al,0
	jne yjw2		push bx
	lea dx,ghh		shl bx,cl
	mov ah,9		cmp bh,9
	int 21h		jbe v1
	jmp qq1		sub bh,9
yjw2:	jmp b1		add bh,40h
b1:	mov bx,ax	v1:	add bh,30h
	call mcat	mm1:	mov aa,bh
	lea dx,ii		pop bx
	mov ah,9		and bl,0fh
	int 21h		push ax
qq1:	mov ah,4ch		push bx
	int 21h		push cx
mcat	proc ;将随机数转换为十六		push dx
	;进制表示的 ASCII 码字符子程序		mov ax,bx
	push ax		mov bl,5
	push bx		div bl
	push cx		cmp ah,3
	push dx		jae vb2
	cmp bx,9		mov www,1
	ja s1		jmp vn2
	push ax	vb2:	mov www,0
	push bx	vn2:	pop dx
	push cx		pop cx
	push dx		pop bx
	mov ax,bx		pop ax
	mov bl,5		cmp bl,9
	div bl		jbe ttl
	cmp ah,3		sub bl,9
	jae vb1		add bl,40h
	mov www,1		jmp rr1
	jmp vn1	ttl:	add bl,30h
vb1:	mov www,0	rr1:	mov aa+1,bl
vn1:	pop dx		mov aa+2,'\$'
	pop cx		lea dx,aa
	pop bx		mov ah,9
	pop ax		int 21h
	add bl,30h	s3:	pop dx
	mov aa,bl		pop cx
	mov aa-1,'\$'		pop bx
	lea dx,aa		pop ax
	mov ah,9		ret
	int 21h		

<pre>mcat endp rand proc push cx push dx push ax sti mov ah,0 ;读时钟计数器的值 int 1ah mov ax,dx ;清高 6 位 and ah,3 mov dl,101 ;除 101,产生 0~100 ;余数 div dl mov bl,ah ;余数存 bx, 作随 ;机数 pop ax pop dx pop cx ret rand endp</pre>	<pre>rand1 proc push cx push dx push ax sti mov ah,0 int 1ah mov ax,cx and ah,3 mov dl,101 div dl mov bl,ah pop ax pop dx pop cx ret rand1 endp code ends end start</pre>
--	--

程序运行结果如图 1-4 所示。

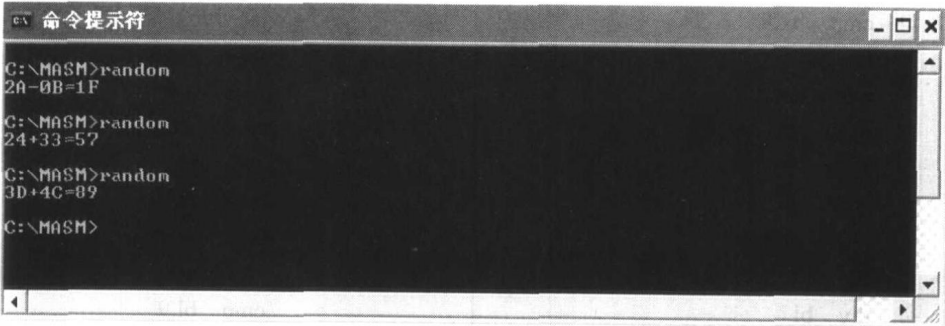


图 1-4 程序运行结果

案例 3 计算平台长度程序

设计内容：已知一个有序数组 a，其中有 n 个元素（n 为整数），a 中相同的元素全部集中在一起，形成一个个“平台”。用汇编语言编程，求出 a 中最大平台的长度。要求：

程序运行时将提示输入有序数组 a，输入完毕后即运算输出 a 中最大平台长度和元素。例如：

运行时从键盘上输入：333 44444 111111

输出结果为：6

111111

设计思路：定义 6 个变量 a、b、a0、b0、a1、b1，其中 a、b、a0、b0 存放从键盘输

入的元素, a、b 为中间变量, a0 始终存放最大平台的元素; a1、b1 存放数组相同元素的个数, a1 中始终存放着最大平台的长度。总的思路是:

用回车键来结束键盘元素的输入。将从键盘输入的第一个元素存入 a、a0 中, 给 a1 赋 1, 接收从键盘输入的第二个元素, 存入 b 中, 判断两个元素是否相同, 若相同, 则 a1 加 1, 否则将第二个元素存入 b0 中, 给 b1 赋 1。接收从键盘输入的第三个元素, 与前一元素比较, 若相同, 则 b1 加 1, 否则, 比较前两次的数组元素长度 a1 与 b1, 若 a1 小于 b1, 则将 b0 的值赋给 a0, b1 的值赋给 a1。再接收下一元素, 重复上述过程, 直到输入回车符为止。程序的流程图如图 1-5 所示。

操作说明: 该程序须在 DOS 下运行, 图 1-6 为程序运行的结果。

源程序 (ptcd.asm) 如下:

```
data segment
obf db '***** welcome you !*****',
0dh,0ah
db 'input data:','$'
a0 db ?
a1 db ?
b0 db ?
b1 db ?
a db ?
b db ?
data ends
stack segment stack
db 200 dup(0)
stack ends
code segment
assume cs:code,ds:data,ss:stack
main: mov ax,data
mov ds,ax
mov ah,9 ;显示提示信息
lea dx,obf
int 21h
inc dx
mov dl,13 ;回车换行
mov ah,2
int 21h
mov dl,10
int 21h
mov ah,1 ;接收键盘输入,
;若为回车键,则转移到 lab1
int 21h ;否则将字符存入
;a 中
cmp al,13
jz lab1
mov a0,a1
```

```
mov a,a1
mov al,1
x1: mov ah,1 ;接收键盘输入,若为
;回车键,则转移到 lab1
int 21h ;否则将字符存入 b 中,
;再将其与前一字符比较,
cmp al,13 ;若不相同,则再将其
;存入 b0、b1 中,若相同,则
jz lab1 ;a1 累加 1
mov b,a1
mov b1,b
cmp b1,a
jnz x2
add a1,1
jmp x1
x2: mov b0,b1
mov b1,1
x3: mov ah,1 ;接收键盘输入,若为
;回车键,则转移到 lab1,输出 a1
int 21h ;a0,再将其与前一字
;符比较,若不相同,则转移到 x4,
mov a,a1 ;若相同,则 b1 累加
cmp b1,a
jnz x4
add b1,1
jmp x3
x4: mov cl,b1
cmp cl,a1 ;比较 a1 与 b1,若 b1
;大,则把 b1 送入 a1, b0 送 a0
jb y1 ;否则转移到 y1
mov cl,b0
mov a0,cl
mov cl,b1
mov a1,cl
```

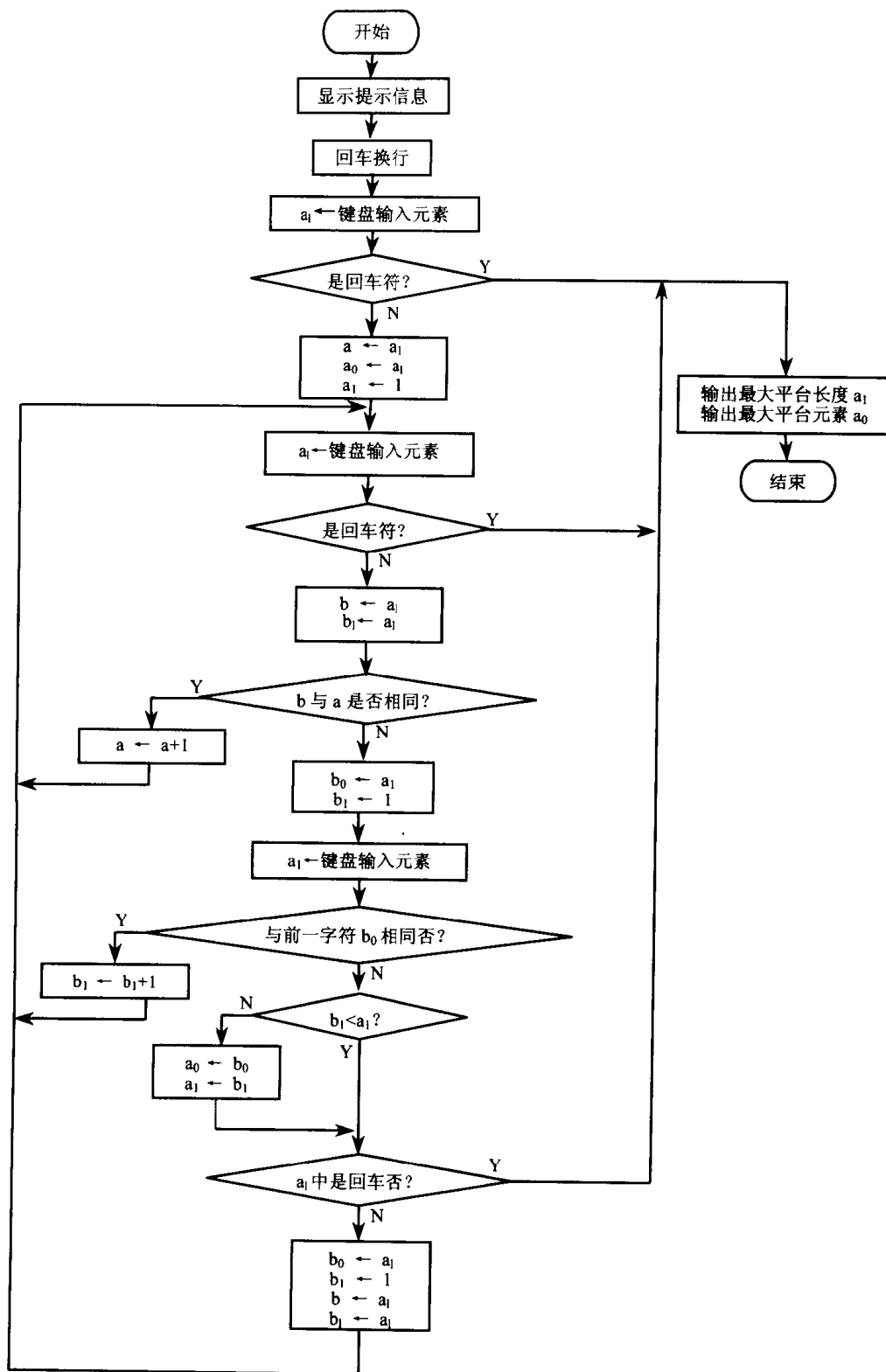


图 1-5 平台长度程序流程图

```

yl:  cmp al,13 ;若输入为回车符, 则
    ;转移到 lab1, 否则返回 x3, 继续
    jz lab1
    mov b0,al
    mov b1,1
    mov b,a1
    mov bl,b
    jmp x3
lab1: mov ah,2 ;输出 a1、a0, 返回
;DOS 系统
    mov dl,13
    int 21h
    mov dl,10
    int 21h
    cmp al,10
    jb z1
    mov ah,0
    mov al,a1
    mov bl,10
    div bl
    add ax,3030h
    mov bx,ax
    mov ah,2
    mov dl,bl

    int 21h
    mov dl,bh
    int 21h
    jmp z2
z1:  add al,30h
    mov dl,a1
    int 21h
    sub al,30h
z2:  mov ah,2
    mov dl,13
    int 21h
    mov dl,10
    int 21h
    mov cx,0
    mov cl,a1
    mov ah,2
m:   mov dl,a0
    int 21h
    loop m
    mov ah,4ch
    int 21h
code ends
end main

```

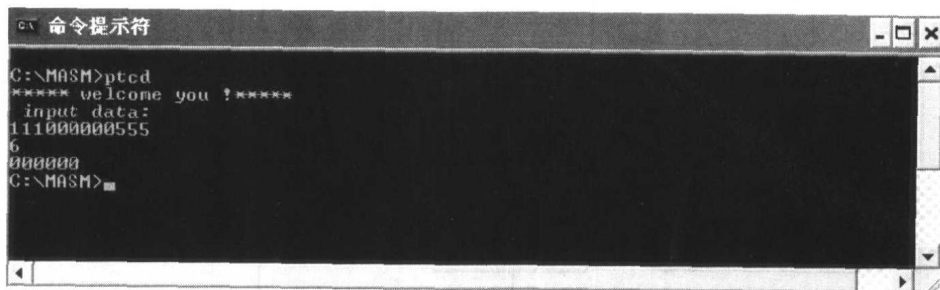


图 1-6 平台长度程序运行结果

案例 4 加减乘除四则计算

设计内容：设计一个能实现加、减、乘、除、取余计算的程序。要求该程序接受从键盘输入的十六进制数，执行相应的计算后，计算结果以十六进制数输出。

设计思路：将键盘输入的表达式预放入一缓冲区中，然后从该缓冲区的第一个字符开始读取判断，当读取到一个有效的表达式时，对表达式进行相应的运算后，输出用十六进制数表示的运算结果；当表达式无效时，输出表达式“error (错误)”的信息或者是有效表达式“JS <Expression>”的提示信息。程序流程图如图 1-7 所示。

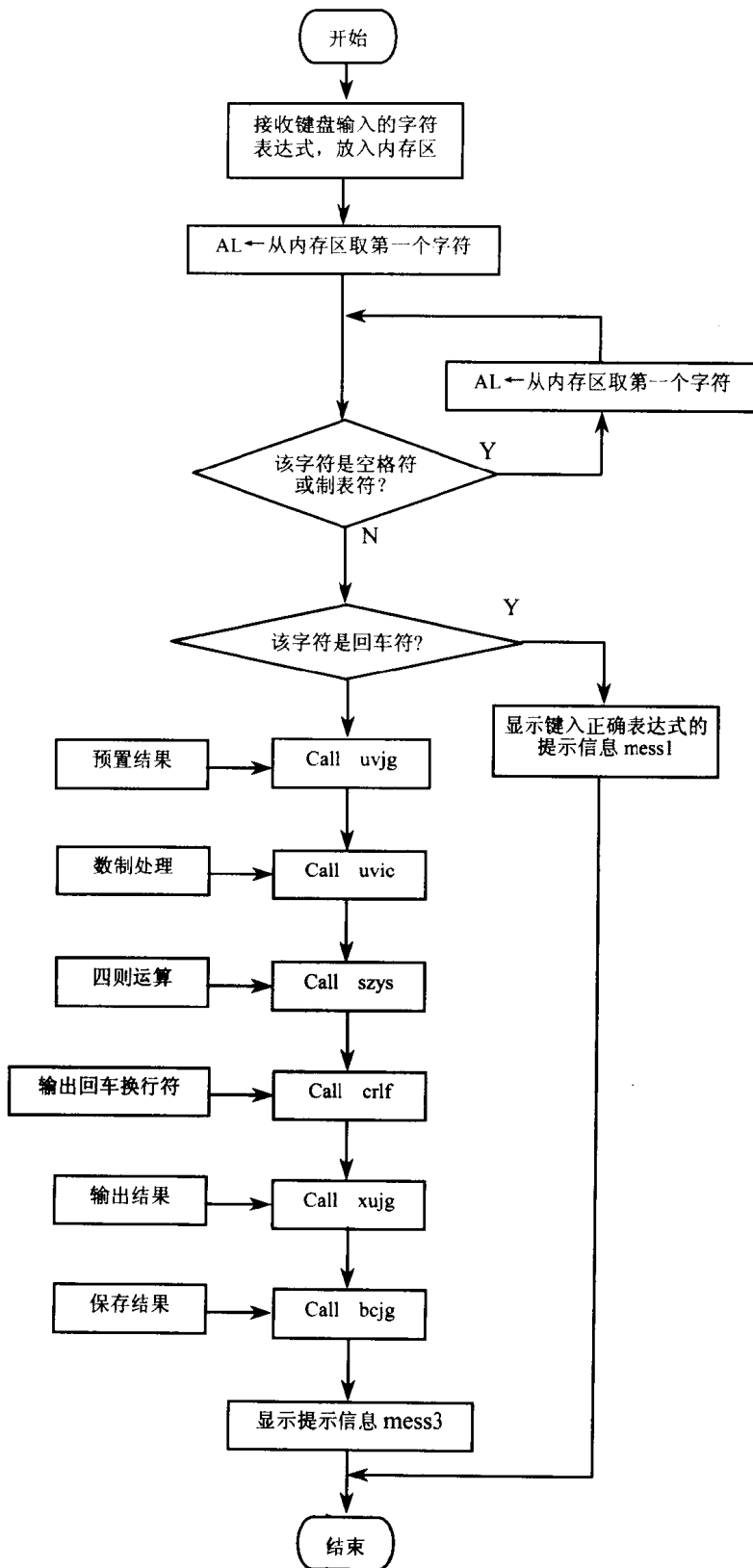


图 1-7 程序结构图

操作说明：该程序须在 DOS 下运行，运行时须在文件名后直接跟上计算表达式，程序运行结果如图 1-8 所示。

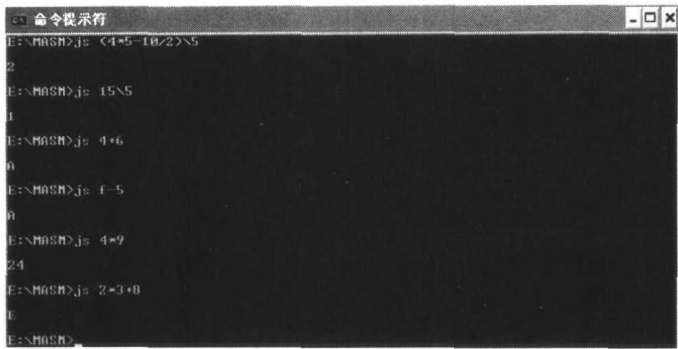


图 1-8 程序运行结果

源程序（js.asm）如下：

b_p	equ byte ptr	code	segment
w_p	equ word ptr		assume cs:code,ds:code
d_p	equ dword ptr		org 100h
;*****主程序开始*****			
newstar:		jgv4	dw 0,0
	jmp star	ysf	label word
	even	ysf1	db 0
num1	dw 0,0	ysf2	db 0
num2	dw 0,0	juv	db 0
num3	dw 0,0	uv	db 0
;*****使 ds:si 指向第一个有效字符*****			
overs	label near		cmp al,9
	cld		jz overs1
	push ax		dec si
overs1:			cmp al,13
	lodsb		pop ax
	cmp al,''		ret
	jz overs1		
;*****输出一个制表符*****			
tab	label near		jmp short disp
	mov al,9		
;*****输出一个回车换行符*****			
crlf	label near		call disp
	mov al,13		mov al,10
;*****显示一个字符*****			
disp	label near		int 21h
	push ax		pop dx
	push dx		pop ax
	mov ah,2		ret
	mov dl,al		