**作业四：乘法指令和过程调用**

1. **输出九九乘法表**

**思路：**

程序的核心是两个嵌套的循环，外层循环控制乘法表的行，内层循环控制每行中的列。每次内层循环都会输出一个乘积，然后输出一个制表符以对齐，最后在外层循环结束时输出一个换行符。

**代码：**

data segment

msg db "The 9\*9 table:", 0ah, 0dh, '$'

num dw ?,?,?,? ;声明一个字（16位）数组num，该数组有四个元素，但初始化为问号?，表示这些值将在程序运行时被填充

res db ?,?,?,? ;声明一个字节（8位）数组

data ends

code segment

assume cs:code, ds:data

start:

mov cx, 9 ;外圈循环次数

mov ax, data ;将数据段的地址存储在寄存器AX中

mov ds, ax ;将数据段寄存器DS设置为数据段的地址，这样程序可以访问数据段中的数据

lea dx, msg ;打印提示语

mov ah, 09h ;将寄存器AH设置为09h，这是DOS的功能号，表示要执行字符串输出

int 21h ;触发DOS中断21h，以执行字符串输出操作，输出提示语

call PrintMultiplication ; 调用过程打印乘法表

mov ah, 4ch

int 21h

PrintMultiplication proc

push cx ;保存外层计数

mov cx, 9 ;外圈循环次数

outer\_loop:

mov [num], cx ;存放乘数

push cx ;保存外层计数

mov cx, 9 ;内圈循环次数

inner\_loop:

mov [num+1], cx ;存放第二个乘数

push cx ;第二个乘数进栈

mov dx, cx ;将第二个乘数加载到寄存器DX中

add dx, 30h ;转换到ASCII

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出第二个乘数

; ... 省略中间代码，与原程序相同 ...

mov dl, 09h ;把水平制表栏移动到DX寄存器低八位

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出水平制表栏

loop inner\_loop ;内层循环结束

mov dx, 0ah ;把换行移动到DX寄存器低八位

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出换行

pop cx ;内层计数

loop outer\_loop ;外层循环结束

pop cx ;还原外层计数

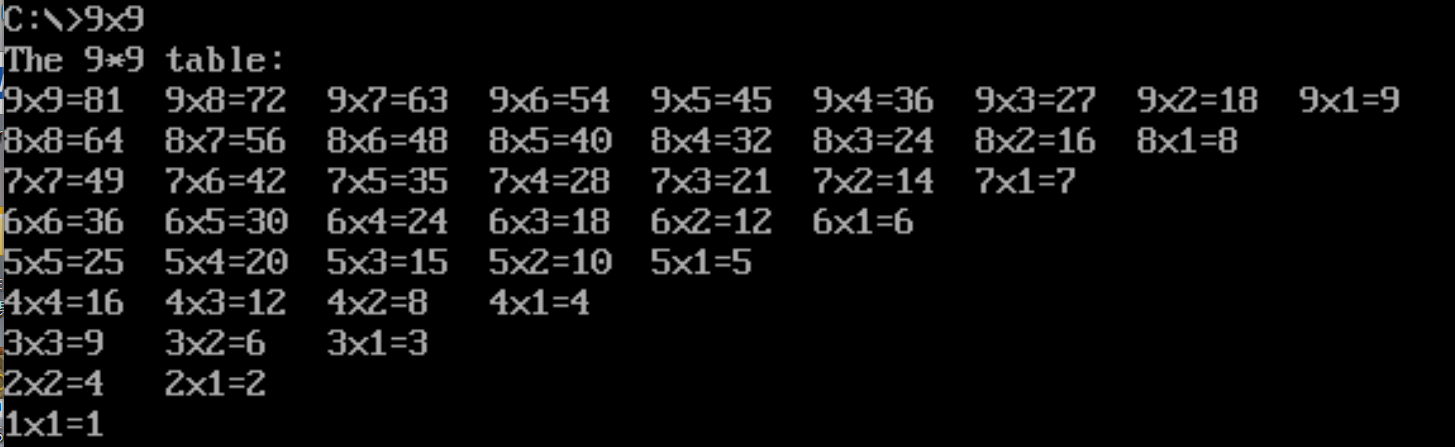
ret

PrintMultiplication endp

code ends

end start

**运行结果：**

****

1. **九九乘法表纠错**

**思路：**

程序通过双重循环遍历9x9乘法表的每一项，并使用MUL指令计算行数和列数的乘积。然后，它将计算结果与table数组中预定义的值进行比较，如果发现任何不匹配，就会输出错误信息。所有比较完成后，程序输出完成信息并结束。

**代码：**

data segment

cnt db 80

table db 7,2,3,4,5,6,7,8,9 ;9\*9表数据

db 2,4,7,8,10,12,14,16,18

db 3,6,9,12,15,18,21,24,27

db 4,8,12,16,7,24,28,32,36

db 5,10,15,20,25,30,35,40,45

db 6,12,18,24,30,7,42,48,54

db 7,14,21,28,35,42,49,56,63

db 8,16,24,32,40,48,56,7,72

db 9,18,27,36,45,54,63,72,81

msg db "x y", 0ah, 0dh, '$'

err db 09h, "error", 0ah, 0dh, '$'

acc db "accomplish!", '$'

data ends

code segment

assume cs:code, ds:data

start:

mov cx, 9 ;行列数

mov ax, data ;将数据段的地址存储在寄存器AX中

mov ds, ax ;将数据段寄存器DS设置为数据段的地址，这样程序可以访问数据段中的数据

lea dx, msg ;打印提示语

mov ah, 09h ;将寄存器AH设置为09h，这是DOS的功能号，表示要执行字符串输出

int 21h ;触发DOS中断21h，以执行字符串输出操作，输出提示语

loop1: ;行循环

push cx ;乘数进栈

mov cx, 9 ;列数

loop2: ;列循环

;计算两数相乘的结果，并比较

mov di, cx ;取当前列数

pop ax ;取当前行数

mov bx, ax ;换个寄存器

push ax ;当前行数再次进栈，在下次列循环中推出再次使用

mul cx ;行数、列数做乘法

mov dl, cnt ;存偏移量

mov si, dx ;8位换16位

cmp al, [table+si];比较行、列乘积与表中数据

je here ;相等跳转，不等打印

;打印行数

add bx, 30h ;行数转化成ASCII码

mov dx, bx ;放入DX寄存器

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出空格

;显示空格

mov dl, 20h ;将DL寄存器设置为' '，表示要输出字符空格

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出空格

;打印列数

add di, 30h ;列数转化成ASCII码

mov dx, di ;放入DX寄存器

mov ah, 02h ;将寄存器AH设置为2，表示要执行字符输出操作

int 21h ;触发DOS中断21h，以输出空格

;打印“error”

lea dx, err ;打印“error”

mov ah, 09h ;将寄存器AH设置为09h，这是DOS的功能号，表示要执行字符串输出

int 21h ;触发DOS中断21h，以执行字符串输出操作，输出提示语

here:

dec cnt ;偏移量自减，下一个数

loop loop2 ;列循环结束

pop cx ;还原行数

loop loop1 ;行循环结束

;打印“accomplish!”

lea dx, acc ;打印“accomplish!”

mov ah, 09h ;将寄存器AH设置为09h，这是DOS的功能号，表示要执行字符串输出

int 21h ;触发DOS中断21h，以执行字符串输出操作，输出提示语

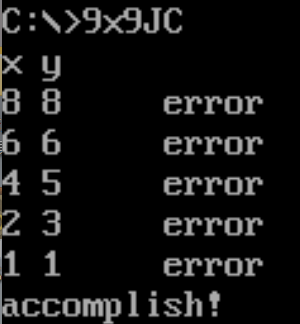
mov ah, 4ch

int 21h

code ends

end start

**运行结果：**

****