

99 乘法表实验报告

2251656 于思源

一、输出 99 乘法表

1. 基本思路

通过两个嵌套循环的方式来打印出一个 9*9 乘法表，其中还包括一个子程序，用于将计算结果从数字转换为 ASCII 码输出。

2. 核心代码

两层循环：

```
.code
START:
    ; 初始化数据段
    MOV AX, @data
    MOV DS, AX

    ; 外层循环 i = 9 到 1
    MOV i, 9
outer_loop:
    CMP i, 0
    JL end_program

    ; 内层循环 j = 1 到 i
    MOV j, 1
inner_loop:
    CMP j, i
    JG new_line

    ; 计算 i * j 并存储结果

    ; 输出 i * j = temp 格式

    INC j
    JMP inner_loop

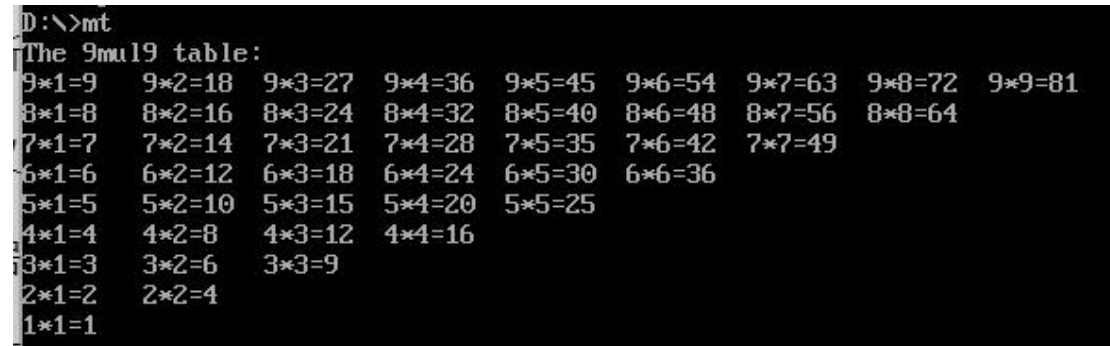
new_line:
    ; 换行并递减 i
    LEA DX, newline
    MOV AH, 09h
    INT 21h
    DEC i
    JMP outer_loop
```

```

end_program:
    ; 结束程序
    MOV AH, 4Ch
    INT 21h

```

3. 结果



```

D:\>mt
The 9mul9 table:
9*1=9  9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
8*1=8  8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64
7*1=7  7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49
6*1=6  6*2=12  6*3=18  6*4=24  6*5=30  6*6=36
5*1=5  5*2=10  5*3=15  5*4=20  5*5=25
4*1=4  4*2=8   4*3=12  4*4=16
3*1=3  3*2=6   3*3=9
2*1=2  2*2=4
1*1=1

```

二、99 乘法表纠错

1. 基本思路

同样是利用两层循环，每次计算对应 99 乘法表的值，并与给出的 99 乘法表进行对比，若发现错误，则输出 `error` 并显示输出乘法的两个操作数。遍历完整个表后，会输出“`accomplish!`”表示检查完成

2. 核心代码

两层循环和比较：

```

MOV x, 1
outer_loop:
    CMP x, 10
    JGE program_end

    ; 内层循环 for (int y = 1; y <= 9; y++)
    MOV y, 1
    inner_loop:
        CMP y, 10
        JGE next_x

        ; 计算 x * y

        ; 获取表中的值 (x-1)*9 + (y-1) 为偏移量

        ; 比较结果是否正确
        CMP AX, result
        JNE print_error

```

```

        ; 继续内层循环
        INC y
        JMP inner_loop

        ; 打印信息
print_error:


        ; 继续内层循环
        INC y
        JMP inner_loop

next_x:
        ; 外层循环递增
        INC x
        JMP outer_loop

获取偏移量:
        MOV AX, x
        DEC AX
        MOV BX, 9
        MUL BX
        ADD AX, y
        DEC AX
        MOV SI, AX
        MOV AL, [table + SI]

```

3. 结果



```

D:\>mtcheck
1 1 error
2 3 error
4 5 error
6 6 error
8 8 error
accomplish!
D:\>

```

三、实验总结

通过本次实验，我深入了解了汇编语言中如何通过循环、条件跳转和子程序调用来实现复杂的功能。在处理乘法表的生成与错误检查过程中，使用了寄存器和内存地址操作，这让我对低级编程中的数据处理有了更加深刻的认识。

在实验过程中，我认识到在处理嵌套循环时，必须精确控制寄存器的增减与条件跳转，否则容易导致循环错误或程序提前结束。

总的来说，这次实验不仅加深了我对汇编语言的理解，也提升了我在实际问题中灵活运用编程技巧的能力。