**99乘法表实验报告**

2251656 于思源

1. **输出99乘法表**
2. **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，用于将计算结构从数字转换为ASCII码输出。

1. **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

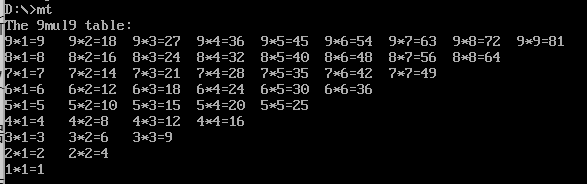
end\_program:

; 结束程序

MOV AH, 4Ch

INT 21h

1. **结果**



1. **99乘法表纠错**
2. **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

1. **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

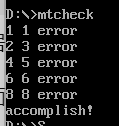
ADD AX, y

DEC AX

MOV SI, AX

MOV AL, [table + SI]

1. **结果**



1. **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。