# 汇编语言课程作业

**————报告4**

2251334 倪朗恩

## 基本要求

1. **输出九九乘法表**

要求至少实现一个过程调用，能正确调用并返回

1. **九九乘法表纠错**

检查9\*9乘法表内数据是否正确，将不正确位置确定下来并显示在屏幕上，数据如下：

TABLE   db 7,2,3,4,5,6,7,8,9

            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

## 输出九九乘法表

### 框架

在main入口函数中使用双循环，同时用AX和BX作为乘数，调用过程CALCU，在CALCU内计算结果并输出？\*？=？ 的基本结构，用AX保存计算结果在过程返回时POP恢复，其中输出AX结果调用过程PRINT\_AX（复用sum代码）

### 具体实现

1. MAIN

输出各种提示信息后，使用双循环调用过程，每次内层循环完成输出换行，每次循环调用一次过程，在过程内集成输出一个乘法的基本功能，简化代码，其中AX保存第一个乘数，BX保存第二个乘数

    MOV CX,[COUNT]

OUTLOOP:

    MOV BX,1

    MOV AX,CX

INTERLOOP:

    CALL CALCU

    INC BX

    CMP BX,CX

    JLE INTERLOOP

1. CALCU

输出AX，“\*”，BX，“=”，然后计算AX\*BX保存至AX，调用过程输出AX（因为位数不确定），最后输出空格

CALCU PROC FAR

    PUSH AX

    PUSH CX

    PUSH DX

    MOV CX,AX

    MOV AH,2

    XOR DX,DX

    MOV DL,AL

    ADD DX,30H

    INT 21H

    MOV DX,"\*"

    INT 21H

    MOV DL,BL

    ADD DX,30H

    INT 21H

    MOV DX,"="

    INT 21H

    MOV AX,CX

    MUL BL

    CALL PRINT\_AX

    XOR DX,DX

    MOV AH,2

    MOV DL,[SPACE]

    INT 21H

    POP DX

    POP CX

    POP AX

    RET

CALCU ENDP

由于改变了AX，CX，DX，所以先PUSH保存最后POP恢复，从而避免影响MAIN中的循环

1. PRINT\_AX

用于输出不定位数的数字，具体原理同sum\_reg中输出部分，先循环除以10确定位数并用BX保存除数，后循环除并输出各位

PRINT\_AX PROC FAR

    PUSH AX

    PUSH BX

    PUSH CX

    PUSH DX

    MOV CX,AX

    MOV BX,1

TRANSF:

    XCHG AX,BX

    MOV DX,10

    MUL DX

    XCHG AX,BX

    DIV BX

    CMP AX,0

    MOV AX,CX

    JNE TRANSF

OUTPUT:

    XCHG AX,BX

    XCHG BX,CX

    MOV BX,10

    XOR DX,DX

    DIV BX

    XCHG BX,CX

    XCHG AX,BX

    XOR DX,DX

    DIV BX

    XCHG AX,DX

    ADD DL,30H

    XCHG AX,CX

    MOV AH,2

    INT 21H

    XCHG AX,CX

    CMP BX,1

    JNE OUTPUT

    POP DX

    POP CX

    POP BX

    POP AX

    RET

PRINT\_AX ENDP

因为AX-DX全被改变，所以先全部保存后回复

## 九九乘法表纠错

### 框架

使用MAIN输出提示信息，使用双重循环对数据逐一进行判断，每次循环先保存AX，后调用CHECK过程，判断对应数据是否等于AX\*BX（九九乘法表），然后在CHECK返回后，根据AX中保存的返回值，判断是否输出错误提示信息，最后恢复AX

### 具体实现

1. MAIN

输出完提示信息后，使用双循环，以AX和BX为乘数，保存AX后，在过程内进行判断，结果是否正确，AX保存返回值TRUE/1或FALSE/0，根据结果判断是否执行/跳过执行错误输出过程

    MOV CX,[COUNT]

    MOV AX,0

OUTLOOP:

    MOV BX,1

    INC AX

INTERLOOP:

    PUSH AX

    CALL CHECK

    CMP AX,1

    POP AX

    JE MAIN\_TRUE

    CALL PRINT\_XY

MAIN\_TRUE:

    INC BX

    CMP BX,[COUNT]

    JBE INTERLOOP

    LOOP OUTLOOP

结合CMP和JMP实现了if判断语句功能

1. CHECK

用于判断TABLE[AX-1][BX-1]是否等于AX\*BX，使用SI和BX访问数据段，用AX返回，所以没有对AX进行保存和恢复，

CHECK PROC FAR

    PUSH BX

    PUSH CX

    PUSH DX

    MOV CX,AX

    MUL BX

    XCHG AX,CX

    MOV DX,AX

    DEC AX

    DEC BX

    MUL [COUNT]

    MOV SI,AX

    MOV AL,[TABLE + BX + SI]

    XCHG AX,DX

    XOR AX,AX

    CMP CX,DX

    JNE FALSE

    INC AX

FALSE:

    POP DX

    POP CX

    POP BX

    RET

CHECK ENDP

结合CMP和JMP改变返回值

1. PRINT\_XY

以AX，BX为位置输入，输出错误提示信息

PRINT\_XY PROC FAR

    PUSH AX

    PUSH CX

    PUSH DX

    MOV DX,AX

    ADD DX,30H

    MOV AH,2

    INT 21H

    MOV DX,[SPACE]

    INT 21H

    MOV DX,BX

    ADD DX,30H

    INT 21H

    MOV CX,3

PRINT\_XY\_LOOP:

    MOV DX,[SPACE]

    INT 21H

    LOOP PRINT\_XY\_LOOP

    MOV AH,9

    LEA DX,ERROR

    INT 21H

    MOV AH,2

    MOV DX,[NEW\_LINE]

    INT 21H

    POP DX

    POP CX

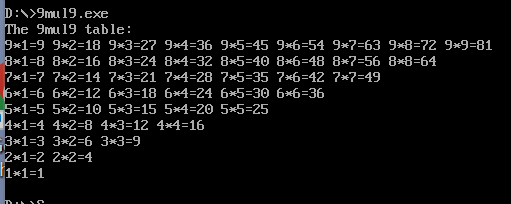
    POP AX

    RET

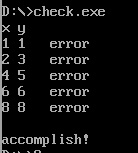
PRINT\_XY ENDP

## 实验结果

1. 输出九九乘法表



1. 九九乘法表纠错



## 问题与心得

1. DEBUG信息

    ;此处必须用AL，不然传进来一个word，之后的比较会出问题

    MOV AL,[TABLE + BX + SI]

    XCHG AX,DX

    ;debug code

    ;PUSH AX

    ;PUSH DX

    ;MOV AH,2

    ;MOV DX,CX

    ;ADD DX,30H

    ;INT 21H

    ;POP DX

    ;POP AX

在代码中必要的地方可以编写一部分debug代码，最后注释掉，有利于检查错误，比如此处使用了AX获取传入参数，导致传入了2字节信息（即TABLE中的两个数被联合传递了），导致了下面的比较报错，于是输出大量错误信息，debug代码可以与debug搭配使用提供更快捷的编程

{不知道之后会不会有类似宏定义的编程便利debug代码编写}

1. 恢复与返回值

可以使用寄存器或者数据段保存返回值（寄存器涉及到过程返回的恢复，目前不会使用），此处使用寄存器，则返回值不用在过程内保存和恢复，而是在外层决定是否保存和恢复，即可实现寄存器返回值

1. LOOP命名

简单的LOOP命名容易出现重复，良好的命名习惯以及加上过程前缀能有效避免重复

1. 赋值类型

要注意传参类型，如心得1中所述，错误的传递可能导致未知异常

1. 选择合适的存储方式

根据具体情形选择数据保存方式能有效减少代码量，比如本次作业中输出九九乘法表的的PRINT\_AX可以使用stack来大量简化代码（本人使用了复杂代码），以及调用过程中大量的保存和恢复

1. 学习了BX和SI共同访问数据段/二维数组的方法