

求和实验报告

2251656 于思源

一、将结果放在寄存器中求和 1-100

1. 基本思路

首先进行初始化，然后使用循环从 1 到 100 进行累加，并将结果存储在 AX 中，最后将累加结果转换为字符串形式并存入缓冲区 buffer。

2. 核心代码

求和：

```
sum_loop:
    ADD AX, CX
    LOOP sum_loop
```

转化为字符串：

```
LEA BX, [buffer+5]
MOV BYTE PTR [BX], '$'
DEC BX

MOV CX, 10

convert_loop:
    XOR DX, DX
    DIV CX
    ADD DL, '0'
    MOV [BX], DL
    DEC BX
    TEST AX, AX
    JNZ convert_loop
```

3. 结果

```
D:\>ADDREG
5050
```

二、将结果放在数据段中求和 1-100

1. 基本思路

与寄存器中相同，区别只是将数据放于数据段中

2. 核心代码

求和：

```
sum_loop:
    ADD [result], CX
    LOOP sum_loop
```

转化为字符串：

```
LEA BX, [buffer+5]
MOV BYTE PTR [BX], '
DEC BX

MOV AX, [result]
MOV CX, 10

convert_loop:
XOR DX, DX
DIV CX
ADD DL, '0'
MOV [BX], DL
DEC BX
TEST AX, AX
JNZ convert_loop
```

3. 结果

```
D:\>ADD5050
5050
```

三、将结果放在栈中求和 1-100

1. 基本思路

同上，区别只是将结果放在栈中。

2. 核心代码

求和：

```
sum_loop:
    ADD AX, CX
    PUSH AX
    LOOP sum_loop
```

输出为字符串：

```
; 从栈中取出最终结果
POP AX

LEA BX, [buffer+5]
MOV BYTE PTR [BX], '$
DEC BX

MOV CX, 10

convert_loop:
XOR DX, DX
DIV CX
ADD DL, '0'
MOV [BX], DL
DEC BX
TEST AX, AX
JNZ convert_loop
```

3. 结果

```
D:\>ADDSTACK
5050
```

四、用户输入 1-100 中数据 num，求和 1-num

1. 基本思路

其他思路同上，此处只给出读取 num 的思路。

逐个字符读取，并将 ASCII 转换为对应数字，并保存在 CL 寄存器中。最后将结果寄存器中的数据*10，再加上 CL 寄存器中的值，持续进行循环，调用 DOS 终端，读取下一个字符，若字符不为回车，则继续读取。

2. 核心代码

读取：

```
read_loop:
    MOV AH, 01h           ; 设置DOS中断功能号01h，用于读取输入
    INT 21h               ; 调用DOS中断，读取一个字符
    SUB AL, '0'           ; 将ASCII字符转换为数字（0-9）

    MOV CL, AL            ; 将输入的数字字符保存到CL中
    MOV AX, BX            ; 将BX中的当前值放入AX中
    MOV BX, 10            ; 将基数10放入BX中
    MUL BX                ; AX = AX * 10（进一位）
    ADD AX, CX            ; AX = AX + CL，将当前输入数字加入结果
    MOV BX, AX            ; 更新BX为新的数字值

    ; 读取下一个字符，直到按下回车
    MOV AH, 01h           ; 设置DOS中断功能号01h
    INT 21h               ; 调用DOS中断，读取下一个字符
    CMP AL, 0Dh           ; 判断是否是回车键（0Dh）
    JNZ read_loop         ; 如果不是回车，则继续读取

    ; 将输入的值放入CX中作为循环次数
    MOV CX, BX            ; 将BX中的输入值放入CX
```

3. 结果

```
D:\>ADDINPUT
Enter a number (1-100): 9
45
```

五、实验总结

通过本次实验，我更深刻地掌握了循环指令和跳转指令的应用。