

Homework 3

1 求和1-100

目前代码还有问题，在处理进制时会出现错误，最后结果输出5151而不是5050。

```
F:\>masm sum
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [sum.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51602 + 464942 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link sum

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [SUM.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

F:\>sum
5151
```

代码如下：

```
.MODEL SMALL
.STACK 100h

.DATA
    resultMessage DB "Sum is: $"
    sumResult DB 6 DUP (0)    ; 存储结果字符串，最多5个数字位+结束符

.CODE
MAIN PROC
    MOV AX, @DATA            ; 初始化数据段
    MOV DS, AX

    ; 初始化寄存器
    XOR CX, CX                ; CX 作为计数器，初始化为 0
    XOR AX, AX                ; AX 作为累加器，初始化为 0
    XOR BX, BX                ; BX 作为计数器

calc_sum:
    INC BX                    ; BX += 1
    ADD AX, BX                ; AX += BX
    CMP BX, 100               ; 如果 BX == 100，结束循环
    JLE calc_sum              ; 如果 BX <= 100，继续循环

    ; 现在 AX 中包含累加和 (5050)
    ; 将累加和转换为字符串并输出
```

```

MOV SI, OFFSET sumResult ; 将 SI 指向 sumResult 缓冲区
CALL Itoa                ; 调用 Itoa 函数将数字转换为字符串

; 输出结果
MOV AH, 09H              ; 调用DOS中断来输出字符串
MOV DX, OFFSET sumResult
INT 21H                  ; DOS中断: 显示字符串

; 退出程序
MOV AH, 4CH              ; DOS中断: 程序结束
INT 21H

MAIN ENDP

; 函数: Itoa
; 将 AX 中的整数转换为 ASCII 并存储在 sumResult 缓冲区中
Itoa PROC
    XOR CX, CX            ; CX 用来计数位数
    MOV BX, 10            ; 将除数设为 10

convert_loop:
    XOR DX, DX            ; 清空 DX
    DIV BX                ; AX = AX / 10, 余数存入 DX
    ADD DL, '0'           ; 将余数转换为 ASCII
    PUSH DX               ; 将数字压栈
    INC CX                ; 记录位数
    CMP AX, 0
    JNE convert_loop      ; 如果 AX 还没被除完, 继续

print_digits:
    POP DX                ; 从栈中取出一个数字
    MOV [SI], DL          ; 将其存入结果缓冲区
    INC SI                ; SI 前移
    LOOP print_digits      ; 循环直到所有数字输出完毕

    MOV BYTE PTR [SI], '$' ; 在字符串末尾加上 '$'
    RET
Itoa ENDP

END MAIN

```

2 反编译c语言方法

```
F:\>debug sumc.exe
-u
076A:0000 0E          PUSH     CS
076A:0001 1F          POP      DS
076A:0002 BA0E00      MOV     DX,000E
076A:0005 B409      MOV     AH,09
076A:0007 CD21      INT     21
076A:0009 BB014C      MOV     BX,4C01
076A:000C CD21      INT     21
076A:000E 54          PUSH     SP
076A:000F 68          DB       68
076A:0010 69          DB       69
076A:0011 7320      JNB     0033
076A:0013 7072      JO      0087
076A:0015 6F          DB       6F
076A:0016 67          DB       67
076A:0017 7261      JB      007A
076A:0019 6D          DB       6D
076A:001A 206361     AND     [BP+DI+61],AH
076A:001D 6E          DB       6E
076A:001E 6E          DB       6E
076A:001F 6F          DB       6F
```

076A:0000	0E	PUSH	CS	； 将代码段寄存器压入堆栈
076A:0001	1F	POP	DS	； 弹出堆栈中的值到数据段寄存器，设置 DS = CS
076A:0002	BA0E00	MOV	DX,000E	； 将 000E 装载到 DX 寄存器
076A:0005	B409	MOV	AH,09	； 将 AH 设为 09，DOS 中断 21H 的功能号：显示字符串
076A:0007	CD21	INT	21	； 调用 DOS 中断 21H 显示字符串
076A:0009	BB014C	MOV	BX,4C01	； 将 4C01 装载到 BX，DOS 中断 21H 的功能号：程序终止
076A:000C	CD21	INT	21	； 调用 DOS 中断 21H 终止程序
076A:000E	54	PUSH	SP	； 将堆栈指针压入堆栈
076A:000F	68	DB	68	； 定义字节 68（可能是数据或无效指令）
076A:0010	69	DB	69	； 定义字节 69（可能是数据或无效指令）
076A:0011	7320	JNB	0033	； 如果无进位则跳转到 0033
076A:0013	7072	JO	0087	； 如果发生溢出则跳转到 0087
076A:0015	6F	DB	6F	； 定义字节 6F（可能是数据或无效指令）
076A:0016	67	DB	67	； 定义字节 67（可能是数据或无效指令）
076A:0017	7261	JB	007A	； 如果有进位则跳转到 007A
076A:0019	6D	DB	6D	； 定义字节 6D（可能是数据或无效指令）
076A:001A	206361	AND	[BP+DI+61],AH	； 对内存地址 [BP+DI+61] 和 AH 进行按位与操作
076A:001D	6E	DB	6E	； 定义字节 6E（可能是数据或无效指令）
076A:001E	6F	DB	6F	； 定义字节 6F（可能是数据或无效指令）