# SUM说明文档

## SUM1-1.asm

用寄存器实现加和

主要分为三步：

1. 寄存器维护计数器，加数，现有的和
2. 循环相加
3. 把16进制的结果转换为10进制输出

详见源码



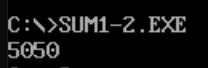
## SUM1-2.asm

用数据段实现加和

基本和1-1一样，就是寄存器变数据段而已

原来存储的数据段变成ds:[0]（直接访问这一段的第一个地址，偏移量0）

详见源码

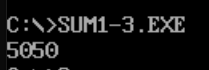


## SUM1-3.asm

用栈实现加和

主要是维护的寄存器变栈，需要初始化一下栈地址和指针，操作上用push和pop

详见源码



## SUM2.asm

主要分为以下步骤：

1、定义字符串常量：

.LC0：定义提示用户输入n的字符串。

.LC1：定义用于格式化输入n的字符串。

.LC2：定义用于输出累加结果的字符串。

2、初始化栈帧和局部变量：

push rbp：保存旧的基址寄存器。

mov rbp, rsp：设置新的基址寄存器为当前栈顶。

sub rsp, 16：为局部变量分配空间。

3、提示用户输入n并读取输入：

使用 printf 输出提示信息。

使用 \_\_isoc99\_scanf 读取用户的输入，并存储到栈上的变量中。

4、累加从1到n的所有数字：

使用 for 循环结构，从1累加到n。

每次循环将当前计数器的值累加到总和变量中，并增加计数器。

5、输出累加结果：

使用 printf 输出累加结果。

6、清理栈帧并返回：

恢复栈帧并返回0表示正常结束。



## SUMdis.asm和SUM.c

c语言输入n，实现1+2+3+…+n，反汇编结果和注释详见源码

