

汇编语言程序设计

# 计数控制循环

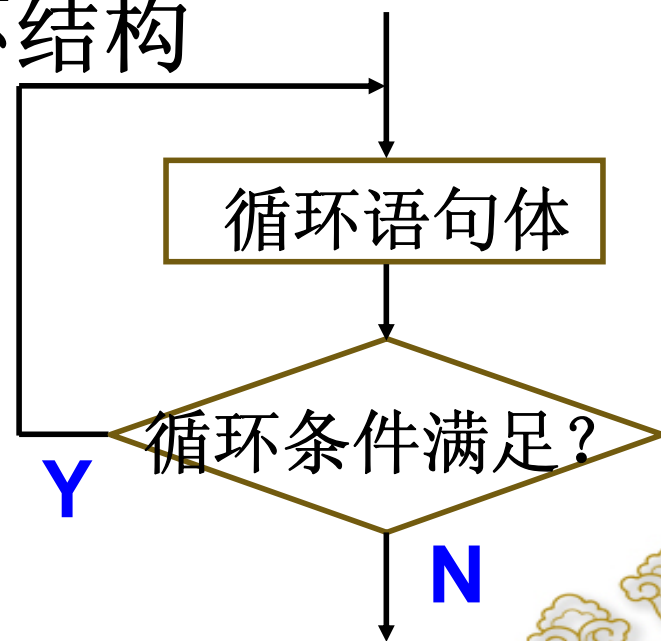


# 计数控制循环

## ➤ 通过次数控制循环

- ▶ 计数可以减量进行，即减到0结束
- ▶ 计数可以增量进行，即达到规定值结束

## ➤ 常见是“先循环、后判断”循环结构



# 数组求和程序

**mov ecx,lengthof array**

**xor eax,eax**

**mov ebx,eax**

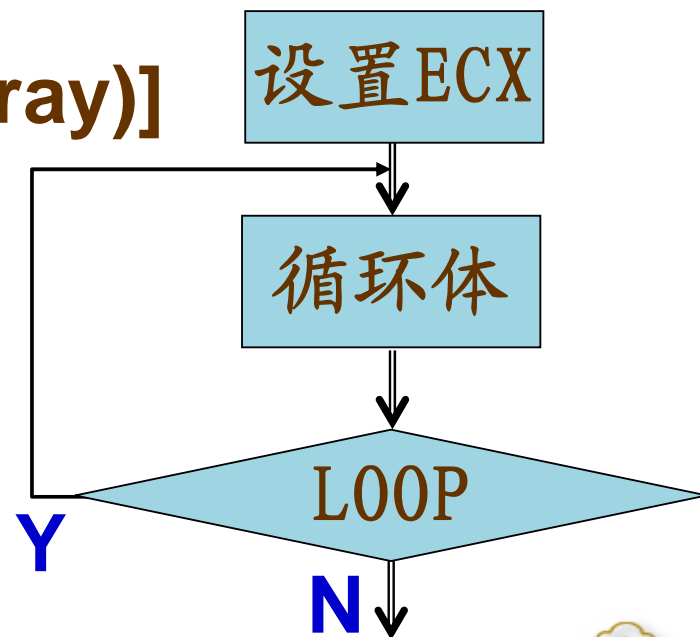
**again:**

**add eax,array[ebx\*(type array)]**

**inc ebx**

**loop again**

**mov sum,eax**



# 求最大值

;数据段

**array dword -3,0,20,900,-56**

**count = lengthof array** ;数组的元素个数

**max dword ?** ;存放最大值

从第1个元素开始逐个比较，保留较大值  
元素个数作为循环次数，实现计数控制  
循环结束，保留值就是最大值



# 求最大值的循环初始化

;代码段

**mov ecx,count-1** ;元素个数减1是循环次数

**mov esi,offset array**

**mov eax,[esi]** ;第一个元素暂存为最大值

ECX作为计数器

ESI寄存器间接寻址访问数组元素



# 循环比较求出最大值

**again:** **add esi,4**

**cmp eax,[esi]** ;与下一个数据比较

**jge next** ;已是较大值，继续

;循环体包含一个单分支程序

**mov eax,[esi]** ;EAX取得更大的数据

**next: loop again**

;计数循环 } 后判断

**mov max,eax** ;保存最大值

先循环



# 使用寄存器相对寻址求最大值程序

**mov ecx,count-1** ;循环次数

**xor esi,esi**

**mov eax,array[esi]** ;EAX保存最大值

**again: add esi,4**

**cmp eax,array[esi]**

**jge next**

**mov eax,array[esi]**

**next: loop again**

**mov max,eax**

;计数循环

;保存最大值

先循环

后判断



# 使用寄存器变址寻址求最大值程序

**mov ecx,count-1** ;循环次数

**xor esi,esi**

**mov eax,array[esi\*4]** ;EAX保存最大值

**again:** **add esi,1**

**cmp eax,array[esi\*4]**

**jge next**

**mov eax,array[esi\*4]**

**next:** **loop again**

**mov max,eax**

;计数循环

;保存最大值

先  
循  
环

后判断





# 增量计数控制循环

**xor esi,esi**

**mov eax,array[esi]** ;EAX保存最大值

**again:** **add esi,1** ;增量计数

**cmp eax,array[esi\*4]**

**jge next**

**mov eax,array[esi\*4]**

**next:** **cmp esi,count-1**

**jb again**

**mov max,eax**

;计数循环

;保存最大值

先循环

后判断

