

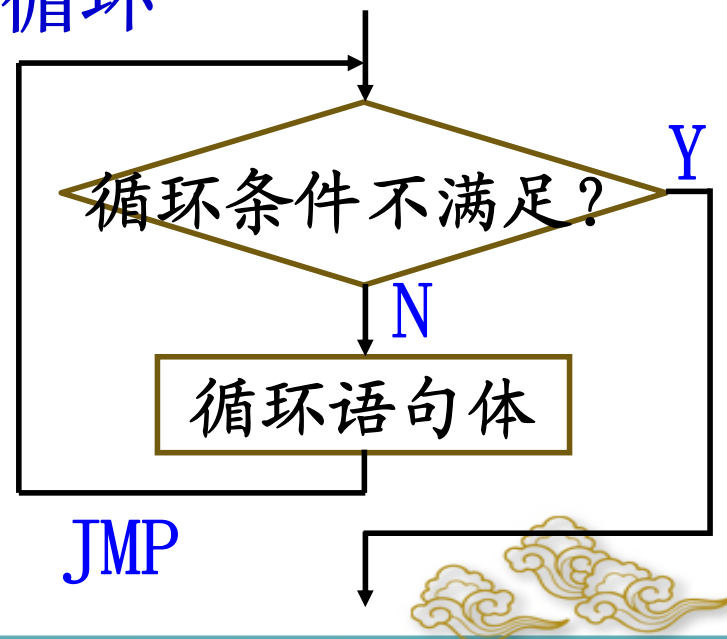
汇编语言程序设计

# 条件控制循环



# 条件控制循环

- 根据条件决定是否进行循环
  - ▶ 使用比较、测试等指令设置状态标志、产生条件
  - ▶ 使用条件转移指令实现循环控制
  - ▶ 常需要使用无条件转移指令配合实现循环
- 多见是“先判断、后循环”结构



# 统计以0结尾的字符串个数

;数据段

**string byte 'Do you have fun?', 0** ;以0结尾的字符串

;代码段

**xor ebx,ebx**

;EBX用于记录字符个数

;也用于指向字符的指针

字符不是0，个数增量1

字符是0，表示字符串结束

字符是否为0就是判断循环是否结束的条件



# 字符个数统计程序

**again: mov al,string[ebx]**

**cmp al,0**

**jz done**

**inc ebx**

**jmp again**

**done: mov eax,ebx**

**call dispuid**

} 先判断

;个数加1

;继续循环

;显示个数

} 后循环

判断字符是0就是条件控制

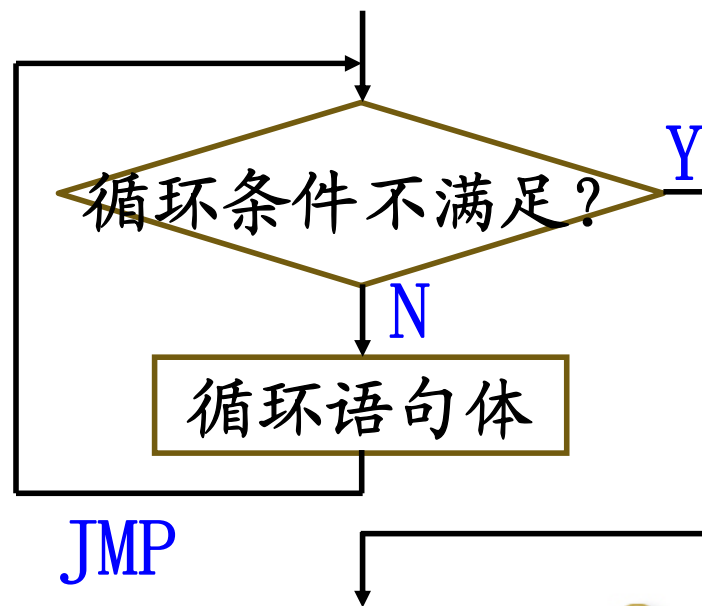


# 先行判断的条件控制循环类似双分支结构

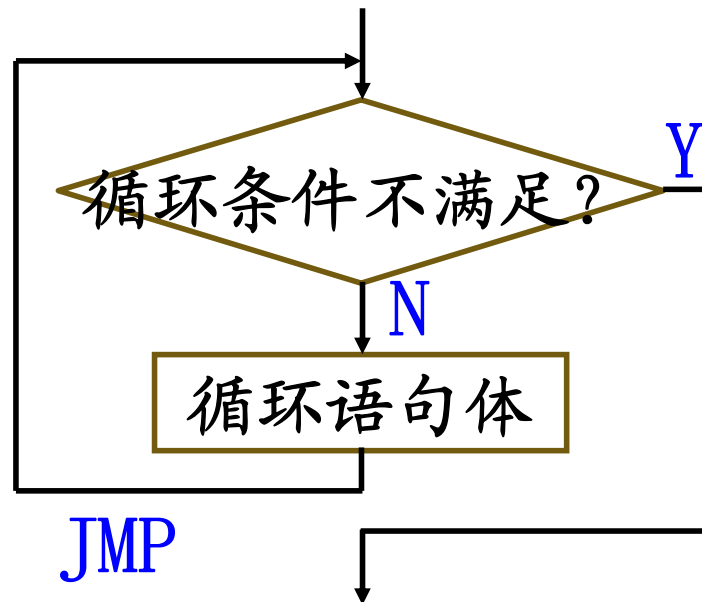
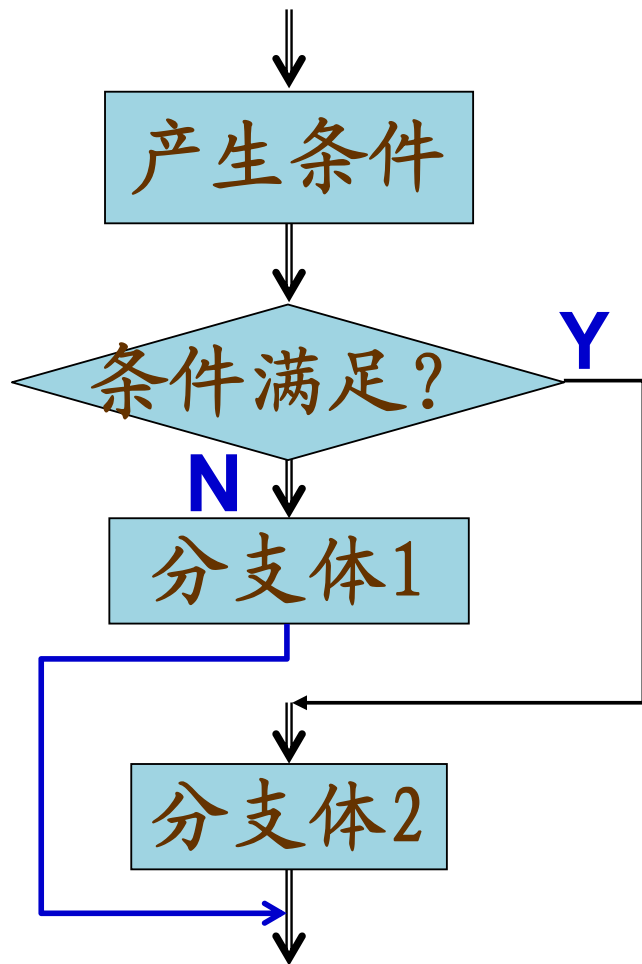
- 顺序执行的分支 → 需要重复执行多次，即循环体
- 另一个分支 → 用于跳出这个循环

```
again:  cmp string[ebx],0  
        jz  done  
        inc ebx  
        jmp again  
done:  ...
```

} 循环



# 双分支结构与“先判断、后循环”结构

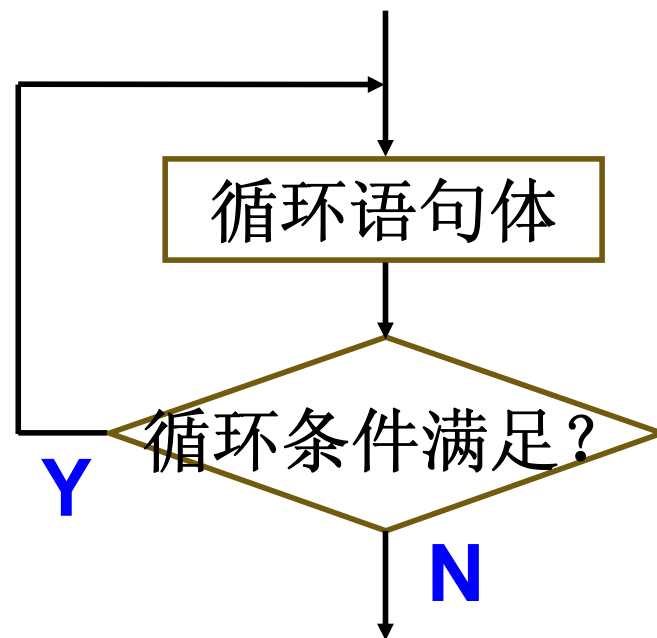


# 先行循环的条件控制循环类似单分支结构

- 分支体 → 就是循环体
- 顺序执行 → 就跳出循环

```
again: inc ebx  
      cmp string[ebx],0  
      jnz again  
      ...           ;退出循环
```

} 循环



# 单分支结构与“先循环、后判断”结构

