```
if (widget_cnt > 0)
{ got_widgets = 1 ; }
else
{ got_widgets = 0 ; }
```

```
; Tell the assembler to allocate space for the variables starting at address
; 002ah. Each variable is initialized to 0.
        org     002ah
widget_cnt:
        db      0
got_widgets:
        db      0

; Tell the assembler that code generation should start at address 0000h.
        org     0000h
; Load widget_cnt & test for 0. The ori instruction sets the condition
; codes. Then jump to the proper instruction sequence.
        lda     widget_cnt
        ori     00h
        jz      no_widgets
        jm      no_widgets
; Set got_widgets to the proper value. We load A with the correct value,
; then store A into got_widgets.
widgets:
        mvi     A,01h
        jmp     set_value
no_widgets:
        mvi     A,00h
set_value:
        sta     got_widgets
```