利用跳转表技术实现根据用户输入的星期几的数字代号在屏幕上显示星期几的英文
2)建立一张跳
名称的功能。
(提示:1)键盘输入的是数字的 ASCII 码,需要转换成数字。
转表,表中存放打印每个星期的程序段的入口地址。

```
.8086
.model small
.stack
.data
   table dw routine_1
         dw routine_2
         dw routine_3
         dw routine_4
         dw routine_5
         dw routine_6
         dw routine_7
   day1 db "monday!", 0Ah,0Dh,'$'
   day2 db "tuesday!", 0Ah,0Dh,'$'
   day3 db "wednesday!", 0Ah,0Dh,'$'
   day4 db "thursday!", 0Ah,0Dh,'$'
   day5 db "friday!", 0Ah,0Dh,'$'
   day6 db "saturday!", 0Ah,0Dh,'$'
   day7 db "sunday!", 0Ah,0Dh,'$'
.code
start:
   mov ax,@data
   mov ds,ax
   mov ah,01h
   int 21h
   sub al,31h
   mov ah,0h
   mov cl,2
   mul cl
   mov si,ax
   mov dl,0ah
   mov ah,02h
   int 21h
   mov dl,0dh
   int 21h
   jmp table[si]

   routine_1:
      mov dx,offset day1
      jmp exit
   routine_2:
      mov dx,offset day2
      jmp exit
   routine_3:
      mov dx,offset day3
      jmp exit
   routine_4:
      mov dx,offset day4
      jmp exit
   routine_5:
      mov dx,offset day5
      jmp exit
   routine_6:
      mov dx,offset day6
      jmp exit
   routine_7:
      mov dx,offset day7
      jmp exit
exit:
   mov ah,09h
   int 21h
   mov ah,4ch
   int 21h
end start
```

STRING BYTE 5 DUP(20H),’$’ ; 20H 为空格的 ASCII 码
请编写完整程序,在屏幕上以十进制的形式将 NUM 这个数打印出来,可以借助 STRING
这个字符串。(NUM 这个数可以定义为一个任意字型数)。

```
.8086
.model small
.stack
.data
   num word 3570h
   string byte 5 dup(20h),'$'
```

```
.code
start:
   mov ax,@data
   mov ds,ax
   mov ax,num
   mov dx,0
   mov bx,offset string+4
   mov cl,10
   mov ch,0
lp: div cx
   add dl,30h
   mov [bx],dl
   mov dl,0
   dec bx
   cmp ax,0
   jnz lp
   inc bx
   mov dx,bx
   mov ah,09h
   int 21h
   mov dl,0ah
   mov ah,02h
   int 21h
   mov dl,0dh
   int 21h
   mov ah,4ch
   int 21h
end start
```

8259 可编程中断

```
.MODEL SMALL
.8086
.stack
.data
come byte 00000000B   ；计算中断到来次数的变量
.code
start:
   mov ax,@data
   mov ds,ax
   cli
   mov ax,0        ; init interrupt vector TABLE
   mov es,ax
   mov si,0        ；由于模拟器的错误，所有中断向量需要指向同一个地址
   mov cx,255
l:  mov ax,offset int0
   mov es:[si],ax
   mov ax,seg int0
   mov es:[si+2],ax
   add si,4
   loop l
   mov ax,offset int0
   mov es:[si],ax
   mov ax,seg int0
   mov es:[si+2],ax
   mov ax,@data
   mov ds,ax
   mov al,00010011b  ;  init  8259
   mov dx,210h
   out dx,al        ; ICW1
   mov al,60h
   mov dx,212h
   out dx,al        ; ICW2
   mov ax,00000001h
   out dx,al        ; ICW4
   mov ax,0
   out dx,al        ; OCW1
   sti
   jmp $
int0  proc
   push ax
   push dx
   push ds
   mov ax,@data
   mov ds,ax
   mov al, come     ；中断到来，计数器+1
   inc al
   mov come, al
   mov al,20h       ；发 EOI 命令
   mov dx,210h
   out dx,al
   pop ds
   pop dx
   pop ax
   iret
int0 endp
END start
```

交通灯

```
.8086
.MODEL SMALL
.stack
.data
   count byte 0h
.code
start:
        mov ax,@data
        mov ds,ax


   mov dx,226h
   mov al,00110111B            ; 8254
   out dx,al
   mov ax,1000h
   mov dx,220h
   out dx,al
   mov al,ah
   out dx,al  ;c0
   mov dx,226h
   mov al,01110111B
   out dx,al
   mov ax,1000h
   mov dx,222h
   out dx,al
   mov al,ah
   out dx,al  ;c1

   mov al,10000000B    ; 8255
   mov dx,206h
   out dx,al
   mov al,10000001B
   mov dx,200h
   out dx,al

   cli
        mov ax,0
        mov es,ax
        mov si,0
        mov cx,255
l:  mov ax,offset int0
   mov es:[si],ax
   mov ax, seg int0
   mov es:[si+2],ax
   add si,4
   loop l
   mov ax,offset int0           ; 修改中断向量表
   mov es:[si],ax
```

```
mov ax,seg int0
mov es:[si+2],ax
mov ax,@data
mov ds,ax
sti



mov al,00010011b  ;  init  8259
mov dx,210h
out dx,al         ; ICW1
;mov al,60h
;mov dx,212h
;out dx,al         ; ICW2
;mov ax,00000001b
;out dx,al         ; ICW4
;mov ax,0
;out dx,al         ; OCW1
MOV AL,00001000B;00001000B
MOV DX,212H
OUT DX,AL
MOV AL,01H
OUT DX,AL
jmp $
```