实验六

一、实验目的：

1．掌握条件转移指令的使用方法；

2．掌握指令对标志寄存器中各标志位的影响情况；

3．掌握标志位和转移指令之间的关系。

二、实验任务：

1．从键盘输入两个一位数，在屏幕上显示他们的积。

2．编写一个程序，判别键盘上输入的字符；若是1-9字符，则显示之；若为A-Z或a-z字符，均显示“C”；若是回车字符<CR>（其ASCII码为0DH），则结束程序，若为其它字符则不显示，继续等待新的字符输入。

3．在屏幕上显示A-Z这26个大写英文字母。

4．编程实现屏幕上输出十进制数字00-29的功能。

三、实验过程：

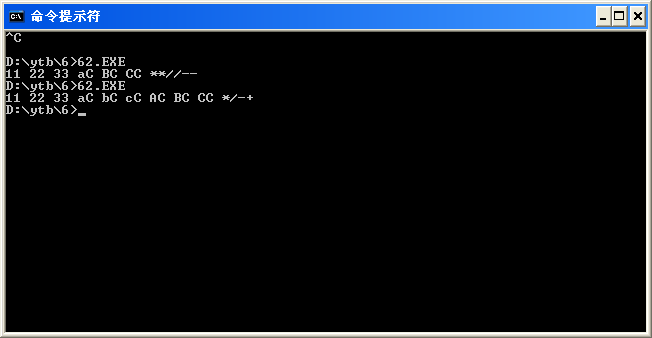
1．程序代码：

|  |
| --- |
| STACK SEGMENT  DB 200 DUP (0)  STACK ENDS  CODE SEGMENT  ASSUME CS: CODE, SS: STACK  BEGIN:  MOV AH, 1  INT 21H  MOV BL, AL ;输入一个字符（第一个一位数），并转存到寄存器BL中  MOV AH, 2  MOV DL, 0DH  INT 21H ;输出回车  MOV AH, 2  MOV DL, 0AH  INT 21H ;输出换行  MOV AH, 1  INT 21H ;输入一个字符（第二个一位数），保存在寄存器AL中  SUB AL, 30H  SUB BL, 30H ;将读入的ASCII码转换成数值  MUL BL ;计算AL和BL的乘积，保存在AX中  MOV BL, 10  DIV BL ;十位数字（商）保存在AL中，个位数字（余数）保存在AH中  ADD AX, 3030H ;个位十位均加上30H，转换成相应的ASCII码  MOV BX, AX  MOV AH, 2  MOV DL, 0DH  INT 21H ;输出回车  MOV AH, 2  MOV DL, 0AH  INT 21H ;输出换行  MOV DL, BL  MOV AH, 2  INT 21H ;输出十位数字  MOV DL, BH  INT 21H ;输出个位数字  MOV AH, 4CH  INT 21H ;程序结束，将控制权交DOS  CODE ENDS  END BEGIN |

2．程序代码：

|  |
| --- |
| ASSUME CS: CODE  CODE segment  START:  MOV AH, 1  INT 21H  CMP AL, 0DH ;判断输入字符是否为回车符  JZ tc ;若为回车符直接退出  CMP AL, '0'  JB NEXT ;如果比'0'小则为其他字符重新开始执行  CMP AL, '9'  JA a ;如果比'9'大跳到a继续判断是否为字母  MOV DL, AL  MOV AH, 2  INT 21H ;显示输入的数字  MOV DL, ' '  INT 21H ;显示空格  JMP START ;重新开始执行  a: CMP AL, 41H  JB NEXT ;如果比'A'小重新开始执行  CMP AL, 5AH  JA b ;如果比'Z'大跳到b继续判断是否为小写字母  c: MOV DL, 'C'  MOV AH, 2  INT 21H ;输出'C'  MOV DL, ' '  INT 21H ;输出空格  NEXT:  JMP START  b: CMP AL, 61H  JB NEXT ;如果比'a'小重新开始执行  CMP AL, 7AH  JA NEXT ;如果比'z'大重新开始执行  JMP c ;输入内容是字母，跳到c继续执行  tc: MOV AH, 4CH  INT 21H ;程序返回  CODE ENDS  END START |

该程序实现了在不同的输入下输出不同内容的功能，演示如下：



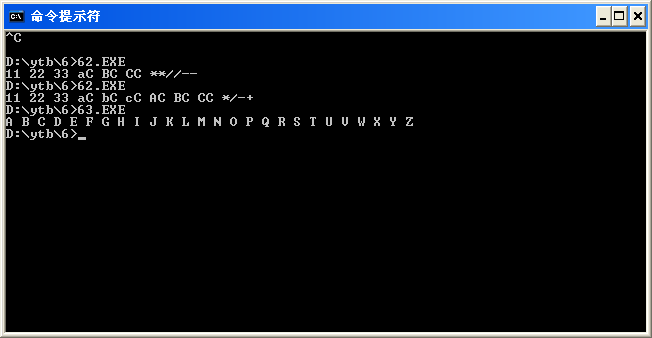
流程图如下（请放大后查看）：



3．程序代码：

|  |
| --- |
| stack segment stack  dw 64 dup (?)  stack ends  code segment  assume cs: code, ss: stack  start:  mov ax, stack  mov ss, ax  mov sp, 64 ;设置栈  mov cx, 001ah ;(cx)=26  mov bl, 41h  mov ah, 02h ;中断21h的功能号2用于输出寄存器dl中的单个字符  a1: mov dl, bl  int 21h ;输出字母  mov dl, ' '  int 21H ;输出空格  inc bl  push cx ;保存寄存器cx的值  mov cx, 0fffh  a2: loop a2 ;延时程序，去抖动  pop cx ;恢复寄存器cx的值  dec cx  jnz a1 ;cx的值不为0时跳转到a1  mov ax, 4c00h  int 21h ;返回DOS  code ends  end start |

该程序运行后会在屏幕上输出A-Z大写的26个英文字母，中间以空格分开：



其流程图如下：



4．程序代码：

|  |
| --- |
| stack segment stack  db 64 dup(?)  stack ends  code segment  assume cs: code, ss: stack  start:  mov dl, 0dh  mov ah, 2  int 21h  mov dl, 0ah  int 21h ;输出换行  mov bl, 0  next:  mov al, bl  mov cl, 4  shr al, cl ;高四位移动到低四位，高四位置0  or al, 30h ;相当于al值增加30h，转换为ASCII码  mov dl, al  mov ah, 2  int 21h ;输出第一位数字  mov dl, bl  and dl, 0fh ;高四位置零  or dl, 30h ;转换为ASCII码  mov ah, 2  int 21h ;输出第二位数字  mov dl, ' '  int 21h ;输出空格  mov al, bl  inc al ;al的值自增  daa ;对al进行加法校正  cmp al, 30h  jnc finish ;(al)≥30时跳到finish  mov bl, al ;bl高、低四位分别设置为要输出的下一个数的十位和个位  mov cx, 0ffffh  delay:  loop delay ;延时程序，去抖动  jmp next ;继续输出下一个数  finish:  mov ah, 4ch  int 21h ;程序返回  code ends  end start |

该程序执行结果如下：

