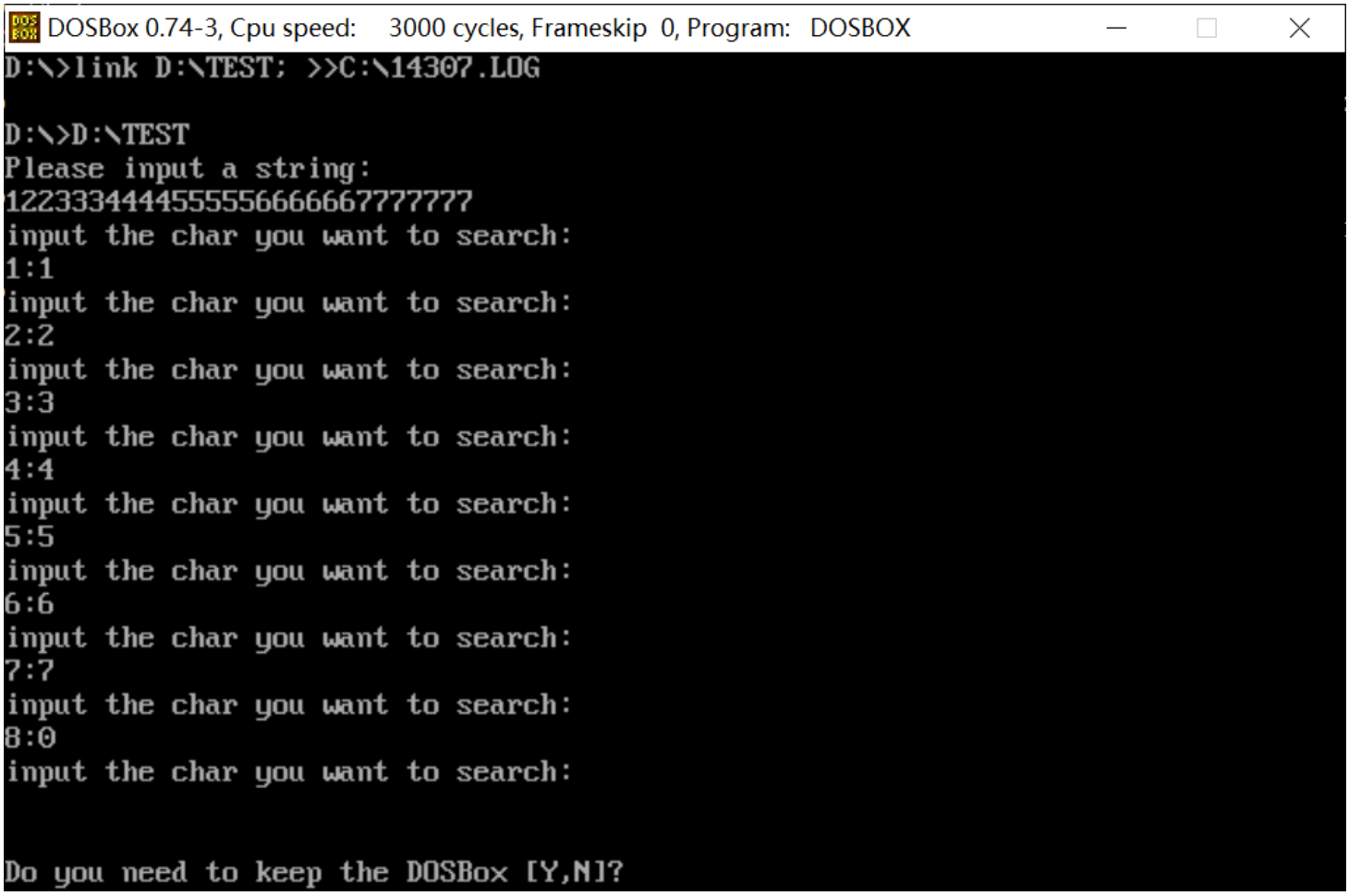
第九周作业

6.8

编写一个有主程序和子程序结构的程序模块。子程序的参数是一个N 字节数组的首地址TABLE，数N 及字符CHAR。要求在N 字节数组中查找字符CHAR，并记录该字符出现的次数。主程序则要求从键盘接收一串字符以建立字节数组TABLE，并逐个显示从键盘输入的每个字符CHAR 以及它在TABLE 数组中出现的次数。(为简化起见，假设出现次数≤15，可以用16 进制形式把它显示出来。)

答：思路如下，首先开辟连续空间存储字符数组，然后对每一个要查找的字符，遍历数组，统计出现次数，最后输出。
代码效果：



代码如下

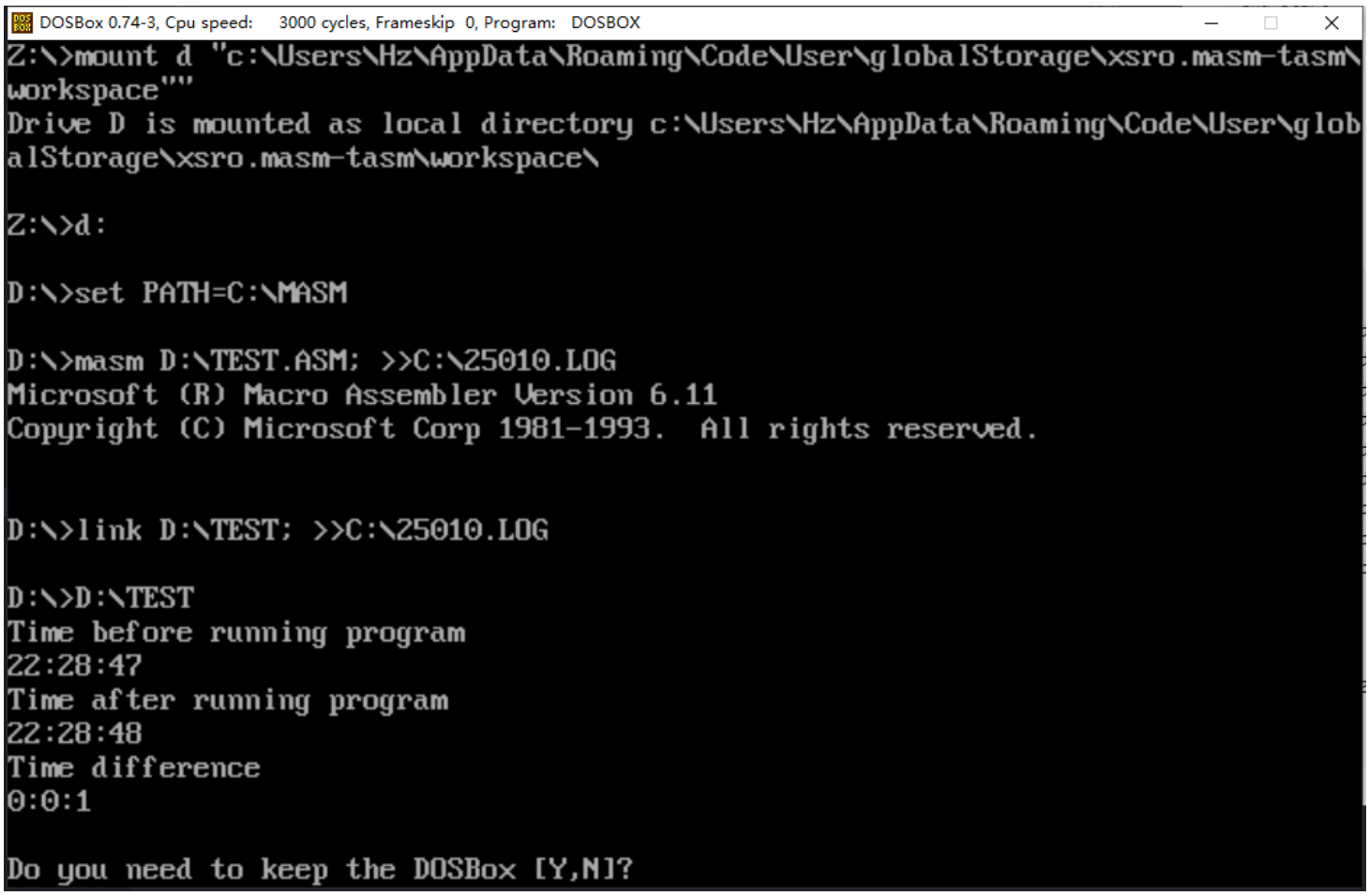
```
data segment
    table    db 100 dup(?)
    nums     dw 0
    char     db ?
    char_num db 0
    change   db 13,10,'$'
    mess1    db 'Please input a string: ',13,10,'$'
    mess2    db 'input the char you want to search: ',13,10,'$'
data ends

code segment
                assume cs:code,ds:data
    start:
                mov    ax,data
                mov    ds,ax

    begin:
                mov    bx,0
                mov    cx,100              ;限定输入的字符数不超过100个
                lea    dx,mess1
                mov    ah,9
                int    21h
    input:
                mov    ah,1
                int    21h

                cmp    al,13               ;输入的字符为回车时，结束输入
                jz     num
                mov    table[bx],al        ;将输入的字符存入table数组中
                inc    bx
                loop   input
    num:
                mov    nums,bx             ;统计输入的字符数
    input_char:                            ;输入要查找的字符
                lea    dx,mess2
                mov    ah,9
                int    21h
    input1:
                mov    ah,1
                int    21h
                cmp    al,13               ;输入的字符为回车时，结束输入
                jz     exit
```

```
                mov     char,al             ;调用搜索子程序查找出现次数
                call    search
                mov     dl,':'
                mov     ah,2
                int     21h

                xor     dl,dl               ;处理输出的数字，转化成16进制ascii码输出
                mov     dl,char_num
                add     dl,'0'
                cmp     dl,'9'
                jle     print
                add     dl,7
    print:
                mov     ah,2
                int     21h
                call    change_line
                jmp     input_char
    exit:
                mov     ah,4ch
                int     21h

search proc near
                mov     si,0
                mov     cx,nums
                mov     char_num,0
                mov     al,char
    loop1:
                cmp     table[si],al
                jne     next
                inc     char_num
    next:
                inc     si
                loop    loop1
                ret
search endp

change_line proc near
                mov     ah,9
                lea     dx,change
                int     21h
                ret
change_line endp
```

```
code ends
end start
```

8.14

试编制一程序，要求测出任一程序的运行时间，并把结果打印出来。
答：思路如下，可以调用dos中的2c号中断，在需要测量的程序前后分别调用，然后将前后的时间差输出即可。
代码效果：



代码如下:

```
data segment
    hour1   db ?
    minute1 db ?
    second1 db ?
    hour2   db ?
    minute2 db ?
    second2 db ?
    h_diff db ?
    m_diff db ?
    s_diff db ?
    mess1 db "Time before running program",0dh,0ah,'$'
    mess2 db "Time after running program",0dh,0ah,'$'
    mess3 db "Time difference",0dh,0ah,'$'
    ctrl db 0dh,0ah,'$'
data ends

code segment
assume cs:code, ds:data
start:
    mov ax,data
    mov ds,ax

    lea dx, mess1
    mov ah, 09h
    int 21h

    call get_time
    mov hour1, ch
    mov minute1, cl
    mov second1, dh

    call display_time1
    call program

    lea dx, mess2
    mov ah, 09h
    int 21h

    call get_time
    mov hour2, ch
    mov minute2, cl
    mov second2, dh
```

```
    call display_time2
    lea dx, mess3
    mov ah, 09h
    int 21h

    mov al, hour2
    sub al, hour1
    mov h_diff, al

    mov al, minute2
    sub al, minute1
    mov m_diff, al

    mov al, second2
    sub al, second1
    mov s_diff, al

    call display_time3
    mov ah,4ch
    int 21h

get_time proc near
    mov ah,2ch
    int 21h
    ret
get_time endp

program proc near
    mov dx,0002fh
ll1:
    mov cx,0ffffh
l1:
    loop l1
    dec dx
    jnz ll1
    ret
program endp

display proc far

                MOV     AH,00H                  ;       清空AH，AH中可能由02，01这样调用int21的残留
                XOR     CX,CX                   ;       CX记录十进制位数
                MOV     BL,10                   ;       除数
```

```
    LOOP1:
                    DIV     BL                      ;           出发操作，余数在AH，商在AL
                    INC     CX                      ;           位数加1
                    PUSH    AX                      ;          入栈保存
                    MOV     AH,00H                  ;       清除余数
                    XOR     AL,00H                  ;       检查是否变为0
                    JNZ     LOOP1                   ;        若还有的除，继续

                    MOV     AH,02H                  ;       AH=02H 输出字符
    LOOP2:
                    POP     DX                      ;
                    MOV     DL,DH                   ;        DH里是要输出的余数
                    ADD     DL,30H                  ;       转ASCII码
                    INT     21H                     ;          输出
                    LOOP    LOOP2                   ;       CX = CX-1 JNZ
                    ret
display endp

display_time1 proc far

                    MOV     AL, hour1
                    call    display

                    mov     ah,02H
                    MOV     DL, ':'                 ; 显示冒号分隔符
                    INT     21h

                    MOV     AL, minute1
                    call    display

                    mov     ah,02H
                    MOV     DL, ':'
                    INT     21h

                    MOV     AL, second1
                    call    display
                    call    change
                    RET
display_time1 endp

display_time2 proc far

                    MOV     AL, hour2
```

```
                    call   display

                    mov    ah,02H
                    MOV    DL, ':'                ; 显示冒号分隔符
                    INT    21h

                    MOV    AL, minute2
                    call   display

                    mov    ah,02H
                    MOV    DL, ':'
                    INT    21h

                    MOV    AL, second2
                    call   display
                    call   change
                    RET
display_time2 endp

display_time3 proc far

                    MOV    AL, h_diff
                    call   display

                    mov    ah,02H
                    MOV    DL, ':'                ; 显示冒号分隔符
                    INT    21h

                    MOV    AL, m_diff
                    call   display

                    mov    ah,02H
                    MOV    DL, ':'
                    INT    21h

                    MOV    AL, s_diff
                    call   display
                    call   change
                    RET
display_time3 endp


change proc near
```

```
    lea dx, ctrl
    mov ah, 09h
    int 21h
    ret
change endp

code ends
end start
```