

4.4

```
DATA SEGMENT
DATA ENDS
```

```
STACK SEGMENT
STACK ENDS
```

```
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK
START:
MOV AH, 1 ;键盘输入字符自动存入 AL 中
INT 21H ;int 中断
SUB AL, 32 ;小写变大写
MOV DL, AL ;赋值给 DX
MOV AH, 2 ;显示输出, DL=输出字符
INT 21H ;int 中断
MOV AH, 4CH ;带返回码结束, AL=返回码
INT 21H
CODE ENDS
END START
```

4.6

```
DATA SEGMENT
BUFY DB ?
BUFZ DB ?
DATA ENDS
```

```
CODE SEGMENT
START:
MOV AX, DATA ;将数据传入 AX 中
MOV DS, AX ;将数据传入 DS 中
MOV AL, BUFY ;先把 bufy 变量传入 al 里
CMP AL, BUFZ ;将 bufy 与 bufz 进行比较
JAE NEXT ;若 AL 大于等于 bufz

MOV AL, BUFY ;若 bufy 大于 bufz, 则更新 AL 内容
NEXT:
MOV BUFZ, AL ;数据处理完成 将更大的数传入 bufz 中
MOV AX, 4C00h ;
INT 21H
CODE ENDS
END START
```

4.23

子程序的参数传递主要有三种方法：1. 用寄存器传递参数；2. 用堆栈传递参数；3. 用存储单元传递参数

1. 用寄存器传递参数

这种方法是将子程序的输入参数由主程序放入规定的寄存器带入子程序中。执行子程序后的结果也放入规定的寄存器带回主程序，采用该方法设计子程序时，主、子程序必须按约定在指定的寄存器中取出或存入指定参数。通常某个寄存器可以用作存放输入参数的寄存器，又同时可以用作存放输出参数的寄存器。

2. 用堆栈传递参数

利用堆栈不仅可用来保存返回地址，而且还可以用来存放主程序和子程序之间传递的参数，这些参数既可以是数据，也可以是地址。用堆栈传递参数的方法是在调用子程序之前，用 PUSH 指令将输入参数压入堆栈，在子程序中通过出栈方式依次获得这些参数。经过子程序操作处理后再将输出参数压入堆栈，返回主程序后再通过出栈获得结果。使用这种方式传递参数时，特别要注意堆栈中断点的保存与恢复。

3. 用存储单元传递参数

还有一种传递较多参数的方法是在内存中使用一个存储区来保存和传递主、子程序间的参数。主程序在调用前将所有输入参数按约定好的次序存入该存储区，进入子程序后按约定从存储区中取出输入参数进行处理，输出参数也按约定大次序存入指定的次序存入指定存储区。返回主程序后就可取得结果。

4.25

子程序的嵌套：嵌套是指一个子程序调用了另一个子程序。在程序执行期间，一个子程序可以调用其他子程序，形成嵌套的调用结构。当一个子程序正在执行时，另一个子程序被调用执行，然后再返回到原始的子程序。

递归：递归是指一个子程序直接或间接地调用自己。递归是一种强大的编程技术，它允许解决某些问题的算法更为简洁。在递归调用中，每个新的调用都创建一个新的栈帧，允许独立的执行。

重入：重入是指一个子程序在被调用的过程中可以被中断，然后再次被调用，而不会产生错误或不一致的结果。具有重入性的子程序可以被多个任务或线程同时调用，而不会相互干扰。这对于多任务操作系统或多线程环境非常重要。、

4.28

```
TO_123 PROC
```

```
    MOV    SI, 0
```

```
LP1:
```

```
    MOV    AH, string[SI]
```

```
    CMP  AH, 0
    JE   EXIT
    CMP  AL, 0
    JNE  EQ1
    OR   AH, 32
    JMP  NEXT
EQ1:
    CMP  AL, 1
    JNE  EQ2
    AND  AH, 32
    JMP  NEXT
EQ2:
    XOR  AH, 32
NEXT:
    MOV  string[SI], AH
    INC  SI
    JMP  LP1
EXIT:
    RET
TO_123 ENDP
```