3-4

data segment

BLOCK DB -2, 5, -113，19，123

COUNT DW $-BLOCK ; COUNT EQU $-BLOCK

MAX DB 0

如上所示，

参考答案

1 定义一个数组，元素为带符号数，求数组元素最大值，存入变量MAX。

data segment

BLOCK DB -2, 2, -3,23

COUNT DW $-BLOCK ; COUNT EQU $-BLOCK

MAX DB 0

ends

stack segment

dw 128 dup(?)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

;my code start here

MOV CX,COUNT

DEC CX ; 设置循环次数

MOV BX, OFFSET BLOCK; 或LEA BX,BLOCK ;

MOV AL, [BX] ;

AGG: INC BX

CMP AL,[BX] ;AL与[SI]比较大小，AL始终存储二者中大的数

JGE NEXT

MOV AL,[BX]

NEXT: LOOP AGG ;

MOV MAX, AL;

;my code end here

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

2定义一个数组，元素为带符号数。编写一个有主程序和子程序结构的程序模块。子程序1的功能是将数组中的元素求绝对值。子程序2的功能是求数组元素的最大值。主程序调用子程序，求数组元素绝对值的最大值，存入某个变量。

参考答案中，子程序直接访问模块中的变量

data segment

BLOCK DB -2, 5, -96，19，23

COUNT DW $-BLOCK ; COUNT EQU $-BLOCK

MAX DB 0

pkey db "press any key...$"

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

; add your code here

CALL ABS\_BLOCK ;调用求绝对值子程序

CALL MAX\_BLOCK ;调用求最大值子程序

HLT

ABS\_BLOCK PROC NEAR ;求绝对值子程序

PUSH SI

PUSH CX

PUSH AX ;保护现场压栈

LEA SI, BLOCK ;取表格有效地址

MOV CX, COUNT ;将元素个数放入循环次数寄存器CX

CHECK: MOV AL, [SI]

OR AL, AL ;(AL)内容不变，置标志

JNS NEXT ;SF=0转NEXT

NEG AL ; SF=1负数，求补，变为正数的补码

NEXT: MOV [SI], AL ;送回数组原始位置

INC SI ;地址增加

DEC CX ;循环次数减少

JNZ CHECK ;CX不为零回到循环

POP AX

POP CX

POP SI ;出栈

RET ;返回主程序

ABS\_BLOCK ENDP

MAX\_BLOCK PROC NEAR ;求最大值子程序

PUSH CX

PUSH BX

PUSH AX ;保护现场压栈

MOV CX,COUNT ;将元素个数放入CX

DEC CX ;比较次数比元素个数少一

MOV BX, OFFSET BLOCK ; 或者LEA BX, BLOCK

MOV AL, [BX] ;把数放入AL

AGG: INC BX ;BX指向下一个元素

CMP AL, [BX] ;比较

JGE THEN ;JUMP IF GREATER OR EQUAL

MOV AL, [BX] ; AL中始终保存较大的那个数

THEN: LOOP AGG ;循环

MOV MAX,AL ;把AL放入MAX中

POP AX

POP BX

POP CX ;出栈

RET ;返回主程序

MAX\_BLOCK ENDP

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

3 编写一个有主程序和子程序结构的程序模块。主程序显示提示信息，如：'To input 10 grades(0~99), separated by Space key, end with ENTER:'

然后从键盘接收10个学生的成绩（0~99）,每个学生成绩之间以空格隔开，建立一个数组，存放到以TABLE 开始的存储器中。子程序实现从中找到最大的数值并存入MAX单元中。主程序调用子程序。

略