5.1 下面的程序执行后，BUF1的前5个字节单元内容分别是多少？

.DATA

DA1 DW 123H, 456H, 0AB00H

DA2 DB 12H, 34H, 56H, 0ABH

DB $-DA2

BUF1 DB 10H DUP(1,2,3)

BUF2 DB 10H, 20H, 30H

.CODE

START:

MOV ESI, OFFSET DA1

MOV EDI, OFFSET BUF1

MOV ECX, 5

LOP: MOV AL, [ESI]

MOV [EDI], AL

INC ESI

INC EDI

LOOP LOP

END START

答：23H, 01H, 56H,04H, 0

5.3 阅读下列程序，回答下列问题：

(1) 该程序完成什么功能?

(2) 程序中的“INC ESI”可否用指令“ADD ESI, 1”代替，为什么?

(3) 程序中的指令LEA ESI, DATA1可用指令\_\_\_\_\_\_\_\_代替。

(4) CLC的目的是\_\_\_\_\_\_\_\_。

.DATA

DATA1 DB 85H, 27H, 4AH

DATA2 DB 93H, 87H, 65H

LEGH EQU DATA2-DATA1

SUM DB 0, 0, 0

.CODE

BEGIN:

LEA ESI, DATA1

LEA EBX, DATA2

LEA EDI, SUM

MOV ECX, LEGH

CLC

AGAIN: MOV AL, [ESI]

ADC AL, [EBX]

MOV [EDI], AL

INC ESI

INC EBX

INC EDI

LOOP AGAIN

END BEGIN

答：（1）将DATA1和DATA2开始的两个24位（3字节）的数相加，结果存入SUM开始的三个字节单元中。

（2）不能用”ADD ESI,1”代替“INC ESI”。因为指令“DC AL, [EBX]”产生的进位标志CF要在下一次循环中被使用，因此ADC指令两次执行的中间其他指令不能影响CF。指令“INC ESI”不影响CF，而指令“ADD ESI,1”会影响CF。

（3）MOV ESI, OFFSET DATA1

（4）使进位标志CF清零，使指令ADC对两个数的最低8位相加。

5.7 在大小为20个字节的数据区中产生排列为下列两种情况之一的数据：

第一种排列：DB ‘0’, ‘A’, ’1’, ’B’……’9’, ’J’

第二种排列：DB ‘A’, ‘0’, ’B’, ’1’……’J’, ’9’

并根据键盘输入1或2决定显示其中一种排列结果。

答：算法分析：数字0的ASCII码为30H，字母A的ASCII码为41H，它们的差值为11H，其他数字和对应的字母ASCII码的差值都是11H。

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO,dwExitCode:DWORD

INCLUDE Irvine32.inc

INCLUDELIB Irvine32.lib

.DATA

FIRST DB 20 DUP(?), 0

SECOND DB 20 DUP(?), 0

.CONST

PROMPT1 DB "输入选择(1或 2):",0AH,0D,0

PROMPT2 DB "显示排列输出："

.CODE

MAIN PROC

LEA ESI, FIRST

LEA EDI,SECOND

MOV ECX, 10

MOV AL, 30H ;将字符‘0’的ASCII码送AL

MOV BL, 41H ;将字符‘A’的ASCII码送BL

LOP:

MOV [ESI], AL

MOV [ESI+1], BL

MOV [EDI],BL

MOV [EDI+1], AL

INC AL ;产生下一个数字符的ASCII码

INC BL ;产生下一个字母的ASCII码

ADD ESI,2

ADD EDI,2

LOOP LOP

LEA EDX, PROMPT1

CALL WriteString

CALL ReadChar

MOV BL, AL

LEA EDX, PROMPT2

CALL WriteString

CMP BL, ‘1’

JZ FIRS

LEA EDX, SECOND

CALL WriteString ;输出第2种排列

JMP NEXT

FIRS:

LEA EDX, FIRST

CALL WriteString ;输出第1种排列

NEXT:

INVOKE ExitProcess, 0

MAIN ENDP

END MAIN

5.11 求数组中10个带符号字节数据中最大值与最小值的差，结果存入字变量RESULT中，要求查找最大值与最小值分别用子程序来完成，并且分别利用寄存器传递参数、堆栈传递参数来实现。

答：

1. 用寄存器传参数

入口参数包括：ECX:数据个数、ESI: 数据首地址

出口参数：AL: 最大值（BL:最小值）

主程序：（省略数据段定义之前的相关语句）

.DATA

ARRAY DB 120, -21, 66, 73, -85, 110, -122, 0, 23, 34

RESULT DB ?

.CODE

MAIN PROC

LEA ESI, ARRAY

MOV ECX, 10

CALL MAX

LEA ESI, ARRAY

MOV ECX, 10

CALL MIN

SUB AL, BL

MOV RESULT, AL

INVOKE ExitProcess, 0

MAIN ENDP

;求最大值子程序

MAX PROC

DEC ECX

MOV AL, [ESI]

LOP:

CMP AL, 1[ESI]

JGE NEXT

MOV AL, 1[ESI]

NEXT:

INC ESI

LOOP LOP

RET

MAX ENDP

;求最小值子程序

MIN PROC

DEC ECX

MOV BL, [ESI]

LOP:

CMP BL, 1[ESI]

JLE NEXT

MOV BL, 1[ESI]

NEXT:

INC ESI

LOOP LOP

RET

MIN ENDP

END MAIN

2. 用堆栈传递参数

**子过程定义时带参数并用USES保护寄存器。**

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO,dwExitCode:DWORD

MAX PROTO, VAR1:DWORD,VAR2:DWORD,VAR3:DWORD

MIN PROTO, VAR1:DWORD,VAR2:DWORD,VAR3:DWORD

;MAX PROTO

;MIN PROTO

INCLUDE Irvine32.inc

INCLUDELIB Irvine32.lib

.DATA

ARRAY SBYTE 120, -21, 66, 73, -85, 110, -122, 1, 23, 34

RESULT BYTE ?

MAXIM SBYTE ?

MINIM SBYTE ?

.CODE

MAIN PROC

LEA ESI, ARRAY

MOV ECX, 10

LEA EDI,MAXIM ;取MAXIM的地址作参数传给MAX过程，作为出口实参。

INVOKE MAX, ESI, ECX, EDI

LEA EDI,MINIM ;取MINIM的地址作参数传给MIN过程，作为出口实参。

INVOKE MIN, ESI,ECX, EDI

MOV DL, MAXIM

SUB DL, MINIM

MOV RESULT, DL

INVOKE ExitProcess, 0

MAIN ENDP

;求最大值子程序

MAX PROC USES ESI ECX EAX, VAR1, VAR2, VAR3

MOV ESI, VAR1

MOV ECX, VAR2

DEC ECX

MOV AL, [ESI]

LOP:

CMP AL, 1[ESI]

JGE NEXT

MOV AL, 1[ESI]

NEXT:

INC ESI

LOOP LOP

MOV ESI,VAR3

MOV [ESI], AL

RET 12

MAX ENDP

;求最小值子程序

MIN PROC USES ESI ECX EAX, VAR1,VAR2,VAR3

MOV ESI, VAR1

MOV ECX, VAR2

DEC ECX

MOV AL, [ESI]

LOP:

CMP AL, 1[ESI]

JLE NEXT

MOV AL, 1[ESI]

NEXT:

INC ESI

LOOP LOP

MOV ESI,VAR3

MOV [ESI], AL

RET 12

MIN ENDP

END MAIN

作业之外的部分习题答案：

5.2 已知BUF开始的数据区中存放有10个大、小写字母，下面程序段完成的功能是什么?

MOV ECX, 10

LEA EBX, BUF

L2: MOV AL, [EBX]

CMP AL, ‘Z’

JBE L1

SUB AL, 20H

MOV [EBX], AL

L1: INC EBX

LOOP L2

答：将其中的小写字母转换为大写字母。说明：小写字母比其对应的大写字母的ASCII码要大20H。

5.4 EDX和EAX联合存放一个64位整数（EDX存放高32位），编写程序将该数据逻辑左移一位。

答：下面仅给出代码段中的主要语句，其他语句省略。也可以只给出SHL和RCL这两条指令。

.code

MAIN PROC

MOV EAX, 12345678H

MOV EDX, 87654321H

SHL EAX, 1

RCL EDX, 1

MAIN ENDP

END MAIN

5.5 试编写程序将一个包含有20个带符号字长度数据的数组ARRAY，分成正整数数组PLUS与负整数数组NEGA两个数组，并且将这两个数组中的数据个数显示出来。

答：

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO,dwExitCode:DWORD

INCLUDE Irvine32.inc

INCLUDELIB Irvine32.lib

.DATA

ARRAY WORD 1234, 4355, 762, -231, 855, -342, 8443, 2234, -430, 984 ,2451, -7123, -6328, 5443, -1120, 9901, -2238,3578,22145, 3287

LEN EQU LENGTHOF ARRAY

PROMPT1 DB "The number of PLUS: ", 0

PROMPT2 DB "The number of NEGA:", 0

PLUS WORD LEN DUP(?) ;预留存放正数的数组

NEGA WORD LEN DUP(?) ;预留存放负数的数组

NUM\_P WORD ? ;存放正数的个数

NUM\_N WORD ? ;存放负数的个数

.CODE

MAIN PROC

MOV ECX, LEN

LEA ESI, ARRAY

LEA EDI, PLUS

LEA EDX, NEGA

LOP:

MOV AX, [ESI]

CMP AX, 0

JL NEGATIVE

MOV [EDI], AX

INC NUM\_P

ADD EDI, 2

JMP NEXT

NEGATIVE:

MOV [EDX], AX

INC NUM\_N

ADD EDX,2

NEXT:ADD ESI,2

LOOP LOP

LEA EDX, PROMPT1

CALL WriteString ;显示提示字符串“The number of PLUS:”

MOV EAX, NUM\_P

CALL WriteDec ;输出正数个数

LEA EDX, PROMPT2

CALL WriteString ;显示提示字符串“The number of NEGA:”

MOV EAX, NUM\_N

CALL WriteDec ;输出负数个数

INVOKE ExitProcess, 0

MAIN ENDP

END MAIN

5.6 试编写程序将32位标志寄存器读出存入一双字存储单元，以字节为单位，高低字节位置交换，即最高字节放最低字节，次高字节放第2字节，依次类推。然后以16进制数格式输出到显示器。

答：

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO,dwExitCode:DWORD

INCLUDE Irvine32.inc

INCLUDELIB Irvine32.lib

.DATA?

FLAG DWORD ?

.CONST

PROMPT DB "输出十六进制数:",0

.CODE

MAIN PROC

PUSHFD ;将32位标志寄存器压栈

POP EAX ;将标志寄存器内容送入EAX

MOV BYTE PTR FLAG+3, AL ;将最低字节存入字单元FLAG的最高字节

SHR EAX, 8 ;将EAX的次低字节移入AL

MOV BYTE PTR FLAG+2, AL ;将次低字节存入字单元FLAG的次高字节

SHR EAX, 8

MOV BYTE PTR FLAG+1, AL

SHR EAX, 8

MOV BYTE PTR FLAG, AL ;将最高字节存入字单元FLAG的最低字节

LEA EDX, PROMPT

CALL WriteString ;显示提示文字

MOV EAX, FLAG

CALL WriteHex ;以十六进制数格式输出

INVOKE ExitProcess, 0

MAIN ENDP

END MAIN