X86 汇编程序设计第二次实验作业

第二次实验,共三道编程题。没有讲解录播。请在本周末前按时提交。

手写程序见文档最后的附件

- 1. 编写一道完整汇编程序,实现冒泡排序,并显示排序前后的结果。 要求(提示:参考讲义例题修改);
 - (1) 建立样本数据区,其中包含两个字(分开,分别由学生本人的 8 位学号的 16 进制字组成: XXXXh,YYYYh)。排序后,这两个字可以分开。
 - (2) 要显示排序前及排序后的字表,每个字中间空一格。
 - (3) 要求将排序、显示内存中的字(十六进制至十进制 ASCII 码)、显示字符、显示字符串等程序块改编为子程序或宏。

```
C:\MASM\BIN\masm 2-1.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [2-1.OBJ]:
Source listing [NUL.LST]: 2-1
Cross-reference [NUL.CRF]: 2-1

50534 + 466010 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\MASM\BIN\link 2-1.obj

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [2-1.EXE]: 2-1
List File [NUL.MAP]: 2-1
Libraries [.LIB]:
C:\MASM\BIN\>
```

```
C:\MASM\BIN>Z-1.exe
12631 4660 22136 5943
4660 5943 12631 22136
C:\MASM\BIN>
```

可以看到,输出了冒泡排序前后的数组(学号水印: 5943=1737H, 12631=3157H),且均没有显示前导 0。

2. 编程实现:从键盘输入一个两位及三位的十进制数,做乘法(假定乘积小于 65535,不考虑溢出),并显示乘法结果的十进制 ASCII 码。

```
:: MASM\BIN>masm 2-2.ASM
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [2-2.OBJ]:
Source listing [NUL.LST]: 2-2
Cross-reference [NUL.CRF]: 2-2
 50650 + 465894 Bytes symbol space free
      0 Warning Errors
      O Severe Errors
C:\MASM\BIN>link 2-2.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983–1987. All rights reserved.
Run File [2-2.EXE]:
List File [NUL.MAP]: 2-2
Libraries [.LIB]:
C:\MASM\BIN>_
C:\MASM\BIN>Z-Z.exe
```

```
C:\MASM\BIN>2-2.exe
17
157
2669
C:\MASM\BIN>_
```

17 * 157 = 2669

- 3. 编程实现 32 位无符号数乘法。
 - (1) 在内存中定义一个无符号数双字 XX, YY, 做乘法,得到一个 64 位的乘积。
 - (2) 显示该乘积的 16 进制 ASCII 码。
 - (3) (选做):显示该乘积的十进制 ASCII 码。

提示: 双字 XX 可拆分成两个字 XXH, XXL; 双字 YY 可拆分成两个字 YYH, YYL; 双字或 64 位结果可用 DD 定义,也可以用 DW 定义,也可以定义为数组(间接寻址)。(XXH, XXL)*(YYH, YYL)=4 个字; 乘法列竖式,注意到处都有进位! 进位加法用 ADC 指令。

```
:: MASM\BIN>masm 2-3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [2-3.0BJ]:
Source listing [NUL.LST]: 2-3
Cross-reference [NUL.CRF]: 2-3
  50458 + 466086 Bytes symbol space free
       0 Warning Errors
0 Severe Errors
C:\MASM\BIN>link 2-3.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983—1987. All rights reserved.
Run File [2-3.EXE]:
List File [NUL.MAP]: 2-3
Libraries [.LIB]:
C:\MASM\BIN>
C:\MASM\BIN>2-3.exe
21682D079C3F90A
150451469471185162
C:\MASM\BIN>
```

XX=17373157H, YY=17061706H, 首先显示乘积的 16 进制, 然后显示 10 进制, 均与使用 Python 计算的结果(下图)一致。

```
>>> x = 0x17061706

>>> y = 0x17373157

>>> x

386275078

>>> y

389493079

>>> x * y

150451469471185162

>>> print('%#x'%(x*y))

0x21682d079c3f90a
```

第一题手写程序:

2-1 冒泡排产	故后是亦	佐果)		M0V	Bx , 0
	F (N) 10 - 1	DADA CTACK		MOV	cx.S
STACK		PARA STACK	DULPZ:	POP	DX
STACK-AREA	DW	INH DUP(?)		CMP	DL, O
STACK-TOP	EUU	4 - STACK-AREA		JNZ	DVLP>-1
STACK	ENDS			CMP	BX.0
DATA	SEGMENT	PARA		J₹	DVLP2-2
TABLE	DW	31574. 12344. 56784. 17374	DULP2-1:	MOV	BX . I
TABLE - LEN	EQU	4		OR	DL. 304
		ин. '¢'		WOV	AH, 2
SPACE	PB	DDH. OAH. '9'		INT	عا ٢
NEW- LINE	DB	01,00.1	DVLP3-2:	LOOP	DVLP2
DATA	EMDS		1	POP	XA
CODE	EGMENT			POP	ВХ
	ASSUME CS	:: CODE . DS ; DATA . SS ; STACK		POP	cx
	PROC	FAR		POP	DX
NIAM	MOV	AX. STACK		RET	
	MOV	ss. AX	DISP_VALUE	ENDP	
	MOV	sp. stack-Top	PISP-ARRAY	Proc	
	MOV	AX. DATA	N72 - 12KM	PUSH D	×
	MOV	DS. AX			x
	CALL	DISP_ARRAY		PUSH B	×
	CALL	SORT		PUSH 1	×
	CALL	DISP_ARRAY		MOV S	1 CX.TABLE_LEN
	MOV.	AX .4C00H		MOV	SI JOST TABLE
	INT	الد	DALP:	MOV P	
	ENDP	-, ,		CALL	DISP-VALUE
MIAM				WOA	DX. OFFSET SPACE
DISP_VALUE	PROC			WOA	AH. 9
	PUSH DX			INL	عالم
	PUSH CX			INC	\$1
	PUSH BX			LOOP	DUTA
	PUSH AX	_		WOA	DX. OFFSET NEWLLINE
	MON CX.			MOV	AH.9
	MOV BX.			TNI	١٢
DVLP1:	XOR DX, D	Λ		POP	5]
	DIN BX			POP	A×
	PUSH DX	v		POP	θx
	roob purb	1		POP	Dx Cx
			DISP-APPAY	ENDP	D

```
2-1
 SORT
             PROC
             PUSH D'
                       DX
                       CX
             PUSH
                       BX
             PUSH
                       AX
             PUSH
                       5]
             PUSH
                       CX , TABLE-LEN
PARTZ:
             MOV
                       CX
             PEC
                       BX.1
42:
             MOV
                       SI . OFFSET TABLE
             MOV
                        CX
             PUSH
                        AX.LSII
             MOV
42-1:
             CMP
                       AX, [5] +2]
                        CONTINUE
             JBE
                        [(+[2]. XA
             XCHG
                       CSII. AX
             MOV
                        BX. O
             MOV
                        5]. 2
CONTINUE:
             APD
             LUBP
                        LP2-1
             POP
                         CX
                         CX
             DEC
                         BX , I
             imp
                         EXIT
             JZ
                         SHORT 42
             JMP
              POP
                          SI
EXI]:
                          AX
              POP
                          BX
              POP
              POP
                          CX
                          DX
              POP
              ENDP
SORT
              ENDS
 CODE
                          NIAM
              EMD
                                                                                         0
```

第二题手写程序:

J-2 *	Fn 1			AMD	AL. OFH	
STACK	SEGMENT	PARA. STACK	w.	XOR	AH. AH	
STACK- AREA		INH. DUP(?)		PUSH	A×	
STACK- TOP	EQU	4- STACK-AREA		MOV	Ax . 51	
STACK	ENPS			MUL	BX AX SI.AX	
		PARA		/*************************************	N	
DATA	SEGMENT	Luku		POP	AX	
ATA	ENDS			ADD	SI. AX	
CODE	SEGMENT			MAD	41	
		DOE, DS: DATA . SS: >TACK	RETURN.	MOV	AX, SI	
11.070/	PROC			POP	BX	
MAIN	•	FAR CTASH		POP	Dx	
	MOV	AA. STACK		POP	SI	
	MOV	55 . AX	GETNOM	EMPP		
	MOU	sp, stack-top	DISP_VALUE	PROC		
	MOV	AX. DATA	17- VNIOC	PUSH	DX	
	NOV	DS. AX		PUSH	CX	
	CALL	GETNUM		PUSH	₽x	
	MOV	BX - AX		PUSH	AX	
	CALL	CETNUM		MOU	CX .2	
	MUL	BX		MON	Bx. lo	
	CALL	DISP. VALUE	DVY1:	XOR	DX. DX	
	MOU	AX , 4000 H		DIV	BX	
	INT	ગામ		PUSH	DX	
NIAM	ENDP			road	DULPI	
GETNUM	PROC			MOV	BX.0	
MEINON	PUSH	SI	Duriba	MOU	cx.S	
	PUSH	DX	DVYZZ	bob	DX	
	PUSH	ВX		TNS	DC.O DVLP2-1	
	3.0			cmp	Bx.o	
	MOV	51.0 BX.10		75	DVLP2-2	
		AX . AX	DVLP2-1:	MOV	Bx . 1	
	XOR		Dec 1.	OR	DL. 30H	
41:	MOV	AH. I		MOV	AH. 2	
	INT	214		INT	711	
	cwb	AL. ODY	DVLP2-2:	LODP	DV4 2	
	JE	RETURN			AX	
	cub	AL. 304		POP	BX	
	JB	LPI		Pop	ΩX CX	
	cmp	AL. 394	DISP-VALUE	ENDP	VA	
	AL	41	CODE	EMDS		
	de la companya della companya della companya de la companya della		W.C	END	NIAM	3

第三题手写程序:

2-5 東北 STACK	、2 けり食 SEGMENT	·para . stack		W0V	BX. WORD PTR[5]+	
STACK-AREA		(WH DUPT?)		MUL	Вх	
	DW Tm:	\$ - STACK_AREA		ADD	CRESULT+>1. AX	
STACK - TOP	ERU	\$ - SINCE TIME!		ADC	CRESULT+4]. DX	
STACK	ENDS			ADC	CRESULT+6]. 0	
DATA	SEGMENT	PARA		WOV	AX . WORD PTR [5].]
XX	DD	1721301 भ		MOV	BX. WORD PTR LD]	+2]
YY	DD	170617064		MUL	Вх	
RESULT	DW	4 DUP(OH)		ADD	[RESUL]+2] . AX	
NEWLINE	DB	ODH, OAH, '4'		ADC	[RESULT+4]. DX	
DATA	ENDS	Salar Annual Control		ADC	[RESUL]+6].0	
DAIN				MOV	AX . NORD PTRESI.	+2]
CODE	SEGMENT			MOV	BX. WAD PTR[D]	+2]
	assume c	s:code. Ds:DATA, ss:STACK		MUL	BX	
NIAM	Proc	FAR		ADD	[RESULT+4]. AX	
	MOV	AX. STACK		ADC	DRESULT+61 DX	
	MOV	SP. STACK_TOP		POP	DX	
	MOV	AX . DATA		202		
	MOU	DS. AX		POP	Bx	
	WOV	MUL -32		POP	AX	
	CALL	DISP - HEX		POP	D]	
	CALL	DX. OFFSET MENLINE		POP	5]	
	WOV	AH. 9		RET		
	INT	21 H	MUL-32	ENDP		
	CALL	DISP-DEC		PROC		
	MOV	AX, 4CW4	DISP_VALUE		Dx	
	INI	214		PUSH		
MAIN	ENDP			PUSH	CX	
MAL				PUSH	ВX	
MUL-32	PROC	\$I		PUSH	A×	
	PUSH	OL		MOV	CX . 4	
	push			WOO	BX.Ib DX.DX	
	push	ΑX	DLP1:	YOR	BX	
	PUSH	BX		DIV	DL. 30H	
	PUSH	DX		add cmp	DL. 384	
	WOV	SI. OFFSET XX		Territory (1)	71.37	
	MOV	DI. OFFET YY		JBE	DL, 'A' - '9' -1	
	MOV	Ax. NORD PTR [5]]	MATT	PUSH	DX.	
	MOV	BX. WRD ITE[D]]	DICIT	600	DLPI	
	WOL	BX [RESULT] - AX		Lov	Def 1	
	VON	[RESULT+>] .DX				(i)

```
DECLP1:
                                                                XOR
                                                                         XQ.XQ
 2-3
                                                                         Ax . [5] +6]
                                                                MOV
                      Cx . 4
              MOV
                                                                         BX
                                                                DIV
DLP2:
                      DX
              POP
                                                                         (s]+6]. Ax
                                                                MOV
                      AH. Z
             MOV
                                                                         Ax . [5]+4]
                                                                MOV
              INT
                      AIL
                                                                         BX
                                                                DIV
             LUOP
                      DLPZ
                                                                         XA.[4+[2]
                                                                MOV
              POP
                      AX
                                                                          [5+[2] . XA
                                                                 MOV
              POP
                       BX
                                                                         BX
                                                                 DIV
                                                                         [5]+2].AX
                      CX
             POP
                                                                 MOV
                                                                          Ax. [S]
                                                                 MOV
              POP
                                                                          BX
                                                                 DIV
DISP_VAWE
              EMPP
                                                                         [S]]. AX
                                                                 MOV
              PROC
                                                                          DX
DISP_HEX
                                                                 PUSH
                       DX
                                                                          DECLPI
              PUSH
                                                                 LUDP
                       CX
              PUSH
                                                                          BX. 0
                                                                 MOV
                       AX
                                                                          CX.20
              PUSH
                                                                 MOV
                       SI
                                                                           DX
              PUSH
                                                                 POP
                                                 DECLP2:
                                                                          DL.O
                       CX · 4
                                                                 CMP
              MOV
                                                                           DECLP 2-1
                       SI. OFFSET RESULT+b
                                                                 JNZ
              MOV
                                                                           BX. D
                                                                 CMP
                       AX.[S]]
              MOV
LI:
                                                                           DECLP2-2
                       JUSP-VALUE
                                                                 JZ
              CALL
                                                                           BX.1
                       51
                                                 DECLP2-1:
                                                                 WOY
              PEC
                                                                 OR
                                                                           DL. 304
                        SI
              DEC
                                                                           A4,2
                                                                 MOV
                        41
              POOD
                                                                 INI
                                                                           214
                        SI
              POP
                                                                 WEP
                                                                            DECLYZ
                                                 DECLP2-2:
                        AX
               POP
                                                                            SI
                                                                 POP
               POP
                        CX
                                                                            DX
                                                                  POP
                        DX
               POP
                                                                            CX
                                                                  POP
DISP_HEX
              ENDP
                                                                             BX
                                                                  POP
              PROC
DISP-DEC
                                                                             AX
                                                                  POP
                       AX
               PUSH
                                                  DISP-DEC
                                                                  EMDP
                        BX
               PUSH
                        CX
               PUSH
                                                                  ENDS
                                                  CODE
               PUSH
                        DX
                                                                  END
                                                                            NIAM
               PUSH
                        51
                        CX. X
               MOV
                        BX. LO
               MOU
                        SI. OFFSET RESULT
               MOV
                                                                                            (1)
```