

# 天津商业大学学生实验报告

开课实验室：403 机房

开课时间 2014 年 5 月 15 日

实验报告 2014 年 5 月 15 日

学院名称	信息工程学院	年级、专业、班	软件工程 1201	学号	20125041	姓名	王靖伟	同组姓名	无
课程名称	计算机组成原理	实验项目名称	实验四 子程序设计			指导教师	高珊		
实验类型	验证 <input type="checkbox"/> 综合 <input type="checkbox"/> 设计 <input type="checkbox"/> 创新 <input type="checkbox"/>						成绩		
教师评语	<div style="text-align: right;">教师签名：_____ 年 月 日</div>								
<p>实验报告内容一般包括以下几个内容：1、目的要求 2、仪器用具及材料（仪器名称及主要规格、用具名称） 3、实验内容及原理（简单但要抓住要点，写出依据原理） 4、操作方法与实验步骤 5、数据图表格（照片） 6、实验过程原始记录 7 数据处理及结果（按实验要求处理数据、结论） 8、作业题 9、讨论（对实验中存在的问题、进一步的想法等进行讨论）</p>									
<p><b>实验报告内容：</b></p> <p><b>一、实验目的</b></p> <p>1、熟悉 Masm for windows 集成实验环境 2012，掌握汇编语言常用指令的使用方法。</p> <p>2、掌握主程序与子程序之间的调用关系及其调用方法。</p> <p>3、掌握子程序调用过程中近程调用与过程调用的区别。</p> <p>4、掌握 Debug 调试程序的基本方法，并能够根据各寄存器的存储结果判断循环结构程序运行的正确性。</p> <p><b>二、实验环境</b></p> <p>Masm for windows 集成实验环境 2012</p> <p><b>三、实验内容</b></p> <p>1、编写汇编程序，对给定数组[10, 20, 30, 5, 60]，求数组元素之和并以十六进制显示。</p> <p>2、利用汇编语言编写找出 10 个无符号数中最大数和最小数，并分别存储。</p> <p><b>四、实验步骤</b></p> <p>1、编写汇编程序，对给定数组[10, 20, 30, 5, 60]，求数组元素之和并以十六进制显示。</p>									

注 1. 每个实验项目一份实验报告。2. 实验报告第一页学生必须使用规定的实验报告纸书写，附页用实验报告附页纸或 A4 纸书写，字迹工整，曲线要画在坐标纸上，线路图要整齐、清楚（不得徒手画）。3. 实验教师必须对每份实验报告进行批改，用红笔指出实验报告中的错、漏之处，并给出评语、成绩，签全名、注明日期。4. 待实验课程结束以后，要求学生把实验报告整理好，交给实验指导教师，加上实验课学生考勤及成绩登记表（见附件 2）、目录和学院统一的封面（见附件 3）后，统一装订成册存档。

# 天津商业大学学生实验报告附页

开课实验室：403 机房

开课时间 2014 年 5 月 15 日

实验报告 2014 年 5 月 15 日

**要求：**求和、十六进制显示输出分别定义为子程序的形式。

其中： 入口参数 SI=数组首地址

CX=数组长度， 出口参数

AX=数组之和

数组定义方法：ARRAY DB 10, 20, 30, 5, 60

2、已知在 DATA 开始的字单元中存放有 10 个无符号整数, 试编制程序挑选出其中的最大数并存放到 MAXV 单元中，挑选出最小数并存放到 MINV 单元中。

**算法分析如下：**

（1）求最大值：10 个数据存储在为 10 个数据的数组中，BX 中存储数组，将首地址数据赋值给 AX, 随后完成指针后移同时数据循环与 AX 比较，如果 AX<[BX], 那么 AX 赋值给 BX，否则指针后移，直到指针指到数组末尾时循环结束，最后将 AX 赋值给 MAXV，然后调用 DOS 输出数据。

（2）求最小值：将首地址数据赋值给 AX, 随后完成指针后移同时数据循环与 AX 比较，如果 AX>[BX], 那么 AX 赋值给 BX，否则指针后移，直到指针指到数组末尾时循环结束，将 AX 赋值给 MAXV，然后调用 DOS 输出数据。

## 五、实验结论

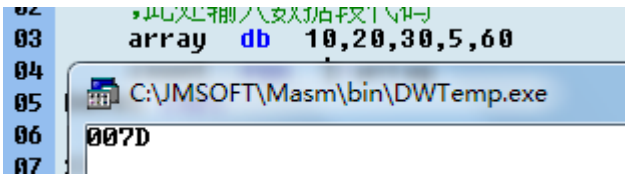


图 1 以十六进制显示数组元素之和

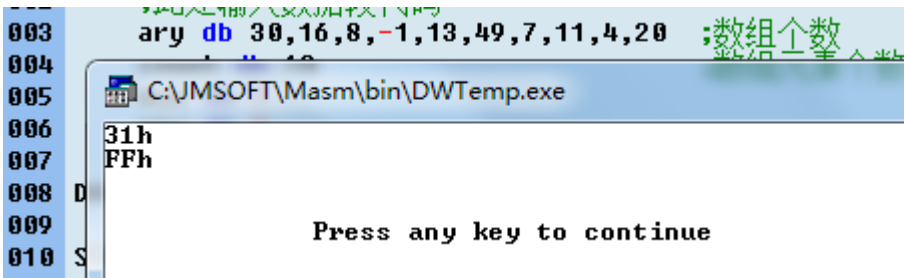


图 2 输出 10 个数中的最大值和最小值

## 六、实验分析

行数	4-1.asm 代码
1	DATAS SEGMENT
2	;此处输入数据段代码
3	array db 10,20,30,5,60

```

4      count equ $-array
5  DATAS ENDS
6
7  STACKS SEGMENT
8      ;此处输入堆栈段代码
9  STACKS ENDS
10
11 CODES SEGMENT
12     ASSUME CS:CODES,DS:DATAS,SS:STACKS
13 START:
14     MOV AX,DATAS
15     MOV DS,AX
16     ;此处输入代码段代码
17     main proc far ;主程序
18         lea si,array
19         mov cx,count
20         call sum1
21         call display
22 out1:MOV AH,4CH
23     INT 21H
24 main endp
25
26 sum1 proc near
27     cmp cx,0
28     jz exit
29     xor ax,ax
30 again:add al,[si]
31     adc ah,0
32     inc si
33     loop again
34 exit: ret
35 sum1 endp
36
37 display proc near
38     mov bx,ax
39     mov ch,4
40 loop1:mov cl,4
41     rol bx,cl
42     mov al,bl
43     and al,0fh
44     or al,30h
45     cmp al,3ah
46     jl print
47     add al,07h
48 print:mov dl,al
49     mov ah,02h
50     int 21h
51     dec ch
52     jnz loop1
53     ret

```

54	display endp
55	
56	CODES ENDS
57	END START

行数	4-2.asm 代码
1	DATAS SEGMENT
2	;此处输入数据段代码
3	ary db 30,16,8,-1,13,49,7,11,4,20 ;数组个数
4	count db 10 ;数组元素个数
5	max db ?
6	min db ?
7	address dw 4 dup(?) ;地址表缓冲区
8	DATAS ENDS
9	
10	STACKS SEGMENT
11	;此处输入堆栈段代码
12	STACKS ENDS
13	
14	CODES SEGMENT
15	ASSUME CS:CODES,DS:DATAS,SS:STACKS
16	START:
17	MOV AX,DATAS
18	MOV DS,AX
19	;此处输入代码段代码
20	mov address,offset ary;
21	mov address+2,offset count
22	mov address+4,offset max
23	mov address+6,offset min
24	lea bx,address ;地址表首地址送入寄存器BX
25	call search ;搜索最大值及最小值
26	mov al,max
27	call outdata ;显示最大值
28	call crlf ;回车换行
29	mov al,min
30	call outdata ;显示最小值
31	mov ah,4ch ;返回DOS
32	int 21h
33	search proc
34	push bx ;现场保护
35	push cx
36	push dx
37	push bp
38	push si
39	push di
40	mov bp,[bx+2] ;数组长度单元地址送入BP
41	mov cx,ds:[bp] ;数组长度-》CX
42	dec cx
43	mov si,[bx+4]

```

44     mov di, [bx+6]
45     mov bp, [bx]
46     mov dl, ds:[bp]
47     mov [si], dl
48     mov [ds], dl
49     inc bp
50  lop:
51     mov dl, ds:[bp]
52     cmp dl, [si]
53     je next
54     jl minc
55     mov [si], dl
56     jmp next
57  minc:
58     cmp dl, [ds]
59     jge next
60     mov [ds], dl
61  next:
62     inc bp
63     loop lop
64     pop di
65     pop si
66     pop bp
67     pop dx
68     pop cx
69     pop bx
70     ret
71  search endp
72  outdata proc near
73     push ax
74     push bx
75     push cx
76     push dx
77     mov bl, al
78     mov cl, 4
79     shr al, cl
80     or al, 30h
81     cmp al, 3ah
82     jb outh
83     add al, 07h
84  outh:
85     mov dl, al
86     mov ah, 02h
87     int 21h
88     mov dl, bl
89     and dl, 0fh
90     or dl, 30h
91     cmp dl, 3ah
92     jb outl
93     add dl, 07h

```

```

94      outl:
95          mov ah,02h
96          int 21h
97          mov dl,'h'
98          mov ah,2
99          int 21h
100         pop dx
101         pop cx
102         pop bx
103         pop ax
104         ret
105     outdata endp
106     crlf proc
107         push ax
108         push dx
109         mov dl,0ah
110         mov ah,2
111         int 21h
112         mov dl,0dh
113         mov ah,2
114         int 21h
115         pop dx
116         pop ax
117         ret
118     crlf endp
119
120     CODES ENDS
121     END START

```