

湖南科技大学计算机科学与工程学院

## 汇编语言 实验报告

专业班级： 计科七班

姓 名： 陈琪琪

学 号： 2102010629

指导教师： 周印明

时 间： 10.09~10.13

地 点： 逸夫楼 418

指导教师评语：

成绩：

等级：

签名：                     

年    月    日

实验名称	实验一：有两个数（字数据）DATA1 和 DATA2，把两数之和放在 DATA3 中，两数之差放在 DATA4 中。		
实验性质 (必修、选修)	必修	实验类型（验证、设计、创新、综合）	设计
实验课时	8	实验日期	2023.10.09
实验仪器设备以及实验软硬件要求	电脑、DosBox、Notepad++		
实验目的	1.熟练掌握汇编语言的程序结构，学会如何使用数据定义、符号定义、模块定义等常用的批示性语句。 2.熟悉汇编语言的工作环境；学会使用汇编语言的编辑、汇编、和连接装配程序的方法；熟悉汇编语言程序的编辑—汇编—连接—运行—调试的过程 3.熟练掌握用 DEBUG 来调试程序，掌握调试程序的反汇编、读写寄存器或存储单元、运行程序等简单的调试命令。 4.熟悉汇编语言指令，学会使用数据传送，算术逻辑运算，处理器控制等简单指令。		

实验内容

一、实验原理

在数据段中定义 DATA1，DATA2，DATA3，DATA4 在代码段中把 DATA1，DATA2 两数相加所得两数之和（用 ADD 指令）送给 DATA3（MOV），再求 DATA1，DATA2 之差（用 SUB 指令），把差送给 DATA4（MOV）。

二、程序流程图

```
graph TD; A([开始]) --> B[求两数之和]; B --> C[和送存储单元]; C --> D[求两数之差]; D --> E[差送存储单元]; E --> F([结束]);
```

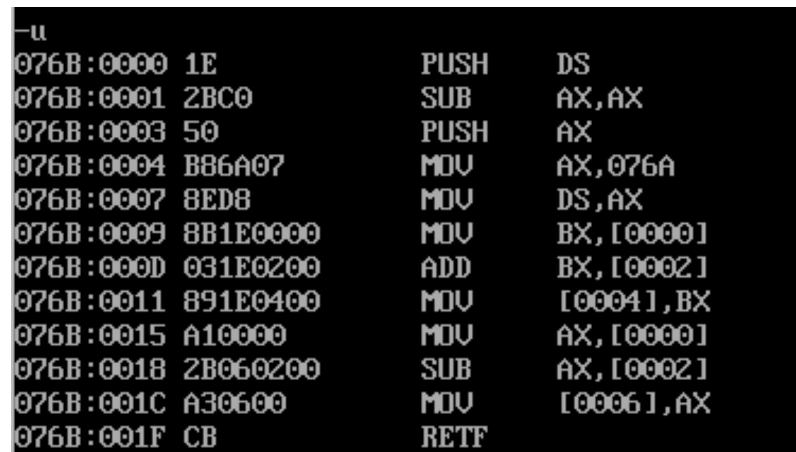
### 三、源程序（附注解）

```
1.      DATA  SEGMENT      ;定义字数据 DATA1, DATA2
2.          DATA1  DW  12H
3.          DATA2  DW  6AH
4.          DATA3  DW  ?
5.          DATA4  DW  ?
6.      DATA  ENDS
7.      CODE  SEGMENT
8.          ASSUME  CS:CODE, DS:DATA
9.      MAIN  PROC  FAR
10.     START:
11.         PUSH  DS
12.         SUB   AX, AX
13.         PUSH  AX          ;DS, 0 入栈
14.         MOV   AX, DATA
15.         MOV   DS, AX      ;设置数据段段地址
16.         MOV   BX, DATA1
17.         ADD   BX, DATA2  ;DATA1+DATA2
18.         MOV   DATA3, BX
19.         MOV   AX, DATA1
20.         SUB   AX, DATA2
21.         MOV   DATA4, AX
22.         RET
23.     MAIN  ENDP
24.     CODE  ENDS
25.     END   START
```

### 实验结果与分析

根据实验题目求两数之和以及两数之差，并根据实验要求进行调试，以下是过程截图以及结果说明。

1.用 U 命令反汇编，了解指令地址



```
-u
076B:0000 1E          PUSH     DS
076B:0001 2BC0        SUB      AX,AX
076B:0003 50          PUSH     AX
076B:0004 B86A07      MOV      AX,076A
076B:0007 8ED8        MOV      DS,AX
076B:0009 8B1E0000    MOV      BX,[0000]
076B:000D 031E0200    ADD      BX,[0002]
076B:0011 891E0400    MOV      [0004],BX
076B:0015 A10000      MOV      AX,[0000]
076B:0018 2B060200    SUB      AX,[0002]
076B:001C A30600      MOV      [0006],AX
076B:001F CB          RETF
```

2.用 G 命令将断点设在 MOV BX, DATA1 处, 再用 D DS: 0 查看数据段的内容

```
-g 0009
AX=076A BX=0000 CX=0030 DX=0000 SP=FFFC BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0009  NU UP EI PL ZR NA PE NC
076B:0009 8B1E0000      MOV     BX,[0000]          DS:0000=000C
-d ds:0
076A:0000 0C 00 6A 00 00 00 00 00-00 00 00 00 00 00 00  ..j.....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 8B 1E 00 00 03 1E 02  .+.P.j.....
076A:0020 00 89 1E 04 00 A1 00 00-2B 06 02 00 A3 06 00 CB  ....+.....
076A:0030 83 C4 06 FF 76 04 E8 95-74 83 C4 02 0B C0 75 08  ...v...t....u.
076A:0040 B8 FF FF 5E 8B E5 5D C3-B8 80 01 50 FF 76 04 E8  ...^..l....P.v..
076A:0050 56 73 83 C4 04 89 46 FC-40 75 20 83 3E 13 02 18  U$....F.@u .>...
076A:0060 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t..v...
```

由图可知, 我们可以观察到在段基址为 076A, 偏移量为 0000 处即为数据代码段存放位置。由于此时数据已经加载到存储中, 且数据定义类型为 DW 以及数据存放形式左侧字节为低字节, 右侧字节为高字节, 所以存放的内容为 0C00 6A00, 其中两个数字代表一个字节。

3.用 G 将断点设在 MOV AX, DATA1 处, 查看 AX, BX 的值,再用 D DS: 0 查看数据段的内容

```
-g 0015
AX=076A BX=0076 CX=0030 DX=0000 SP=FFFC BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0015  NU UP EI PL NZ AC PD NC
076B:0015 A10000      MOV     AX,[0000]          DS:0000=000C
-d ds:0
076A:0000 0C 00 6A 00 76 00 00 00-00 00 00 00 00 00 00  ..j.v.....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 8B 1E 00 00 03 1E 02  .+.P.j.....
076A:0020 00 89 1E 04 00 A1 00 00-2B 06 02 00 A3 06 00 CB  ....+.....
076A:0030 83 C4 06 FF 76 04 E8 95-74 83 C4 02 0B C0 75 08  ...v...t....u.
076A:0040 B8 FF FF 5E 8B E5 5D C3-B8 80 01 50 FF 76 04 E8  ...^..l....P.v..
076A:0050 56 73 83 C4 04 89 46 FC-40 75 20 83 3E 13 02 18  U$....F.@u .>...
076A:0060 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t..v...
```

由图可知, 此时程序已经处理完两数之和并把结果通过 BX 存储在 DATA3 中, 而 DATA3 也是定义在数据段且定义类型为 DW 的, 所以占位 2 个字节, 且以 16 进制形式进行存储, 所以内容呈现为 7600 紧跟前面两数之后。

4.用 G 将断点设在 RET 处, 查看 AX, BX 的值,再用 D DS: 0 查看数据段的内容

```
-g 001f
AX=FFA2 BX=0076 CX=0030 DX=0000 SP=FFFC BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001F  NU UP EI NG NZ NA PD CY
076B:001F CB      RETF
-d ds:0
076A:0000 0C 00 6A 00 76 00 A2 FF-00 00 00 00 00 00 00  ..j.v.....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 8B 1E 00 00 03 1E 02  .+.P.j.....
076A:0020 00 89 1E 04 00 A1 00 00-2B 06 02 00 A3 06 00 CB  ....+.....
076A:0030 83 C4 06 FF 76 04 E8 95-74 83 C4 02 0B C0 75 08  ...v...t....u.
076A:0040 B8 FF FF 5E 8B E5 5D CC-B8 80 01 50 FF 76 04 E8  ...^..l....P.v..
076A:0050 56 73 83 C4 04 89 46 FC-40 75 20 83 3E 13 02 18  U$....F.@u .>...
076A:0060 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t..v...
```

由图可知, 此时程序运行至结尾处, 对于两数之和以及两数之差都已经运算完毕。两数之差通过 AX 存储在 DATA4 中, 而 DATA4 也是定义在数据段且定义类型为 DW 的, 所以占位 2 个字节, 且以 16 进制补码形式进行存储, 所以内容呈现为 FFA2 紧跟前面三数之后。

实验名称	实验二：数据段中有一个字符串 DATA1，串长为 3，内容为‘CDE’，还有另一个字符串为 DATA2，长度为 4，内容为‘ABCD’。要求在附加段中的字符串 DATA3 中得到‘ABDCDE’。		
实验性质 (必修、选修)	必修	实验类型（验证、设计、创新、综合）	设计
实验课时	8	实验日期	2023.10.09
实验仪器设备以及实验软硬件要求	电脑、DosBox、Notepad++		
实验目的	1.熟练掌握汇编语言的程序结构，学会如何使用数据定义、符号定义、模块定义等常用的批示性语句。 2.熟悉汇编语言的工作环境；学会使用汇编语言的编辑、汇编、和连接装配程序的方法；熟悉汇编语言程序的编辑—汇编—连接—运行—调试的过程 3.熟练掌握用 DEBUG 来调试程序，掌握调试程序的反汇编、读写寄存器或存储单元、运行程序等简单的调试命令。 4.熟悉汇编语言指令，学会使用数据传送，串处理指令，处理器控制等简单指令。		

实验内容

一、实验原理

先拷贝 data2 的数据到 data3 中，然后拷贝 data1 的数据到 data3 中，最后更换段基址并输出运行结果即可。此过程因为是使用偏移量进行拷贝的，所以需要使用 loop 语句。

二、程序流程图

```
graph TD; Start([开始]) --> Step1[拷贝data2到data3]; Step1 --> Step2[拷贝data1到data3]; Step2 --> Step3[更换段基址]; Step3 --> Step4[输出data3]; Step4 --> End([结束]);
```

### 三、源程序（附注解）

```
1.  data segment
2.      data1 db 'CDE'
3.      data2 db 'ABCD'
4.  data ends
5.
6.  esdata segment
7.      data3 db 7 dup(?),'$'
8.  esdata ends
9.
10. code segment
11.  assume cs:code,ds:data,es:esdata
12.  start:
13.      mov ax,data ;送段地址
14.      mov ds,ax
15.      mov ax,esdata
16.      mov es,ax
17.      mov si,offset data1 ;取变量的偏移量
18.      mov bx,offset data2
19.      mov di,offset data3
20.      mov cx,4 ;把 ABCD 送到 data3 前 4 个单元
21.  loop1:
22.      mov al,[bx]
23.      mov es:[di],al
24.      inc bx
25.      inc di
26.      loop loop1
27.      mov cx,3 ;把 CDE 送到 data3 后 3 个单元
28.  loop2:
29.      mov al,[si]
30.      mov es:[di],al
31.      inc si
32.      inc di
33.      loop loop2
34.      mov bx,es
35.      mov ds,bx
36.      lea dx,data3
37.      mov ah,09H
38.      int 21h
39.      mov ah,4ch ;退出
40.      int 21h
41.  code ends
42.  end start
```

## 实验结果与分析

根据实验题目拼接字符串，并根据实验要求进行调试，以下是过程截图以及结果说明。  
运行结果：

```
C:\>test2.exe
ABCD CDE
```

### 1. 用 U 命令反汇编，了解指令地址

```
C:\>debug test2.exe
-u
076C:0000 B86A07      MOV     AX,076A
076C:0003 8ED8           MOV     DS,AX
076C:0005 B86B07      MOV     AX,076B
076C:0008 BEC0           MOV     ES,AX
076C:000A BE0000      MOV     SI,0000
076C:000D BB0300      MOV     BX,0003
076C:0010 BF0000      MOV     DI,0000
076C:0013 B90400      MOV     CX,0004
076C:0016 8A07           MOV     AL,[BX]
076C:0018 26            ES:
076C:0019 8805           MOV     [DI],AL
076C:001B 43            INC     BX
076C:001C 47            INC     DI
076C:001D E2F7           LOOP    0016
076C:001F B90300      MOV     CX,0003
```

### 2. 查看数据初始状态

```
-g 0010
AX=076B BX=0003 CX=0060 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=076B SS=0769 CS=076C IP=0010  NU UP EI PL NZ NA PO NC
076C:0010 BF0000      MOV     DI,0000
-D DS:0
076A:0000 43 44 45 41 42 43 44 00-00 00 00 00 00 00 00 00  CDEABCD.....
076A:0010 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
076A:0020 B8 6A 07 8E D8 B8 6B 07-8E C0 BB 03 00 BE 00 00  .j....k.....
076A:0030 BF 00 00 B9 04 00 8A 07-26 88 05 43 47 E2 F7 B9  .....&..CG...
076A:0040 03 00 8A 04 26 88 05 46-47 E2 F7 B0 24 26 88 05  ....&..FG...$&..
076A:0050 8C C3 8E DB 8D 16 00 00-B4 09 CD 21 B4 4C CD 21  .....!.L.!
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .eP.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s.....
```

由图可知，由于 data1 内容为 CDE，data2 内容为 ABCD，且它们的定义类型均为 DB，占字节 1 位，即 2 个数字，且存储形式是 16 进制 ASCII 值，所以在数据段基址 076A，偏移量为 0000 处的存储内容为 43 44 45 41 42 43 44。

### 3. 查看第一个循环后的存储结果，即将 data2 拷贝至 data3 处的结果

```
-g 001f
AX=0744 BX=0007 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0004
DS=076A ES=076B SS=0769 CS=076C IP=001F  NU UP EI PL NZ NA PO NC
076C:001F B90300      MOV     CX,0003
-D DS:0
076A:0000 43 44 45 41 42 43 44 00-00 00 00 00 00 00 00  CDEABCD.....
076A:0010 41 42 43 44 00 00 00 00 00-00 00 00 00 00 00 00  ABCD.....
076A:0020 B8 6A 07 8E D8 B8 6B 07-8E C0 BB 03 00 BE 00 00  .j....k.....
076A:0030 BF 00 00 B9 04 00 8A 07-26 88 05 43 47 E2 F7 B9  .....&..CG...
076A:0040 03 00 8A 04 26 88 05 46-47 E2 F7 B0 24 26 88 05  ....&..FG...$&..
076A:0050 8C C3 8E DB 8D 16 00 00-B4 09 CD 21 B4 4C CD 21  .....!.L.!
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .eP.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s.....
```

由图可知，data3 所在段为 ES，偏移量为 0010，且通过循环将 data2 进行拷贝，从而存放数据为 ABCD，对应 16 进制 ASCII 码值为 41 42 43 44。

4.查看第二个循环后的存储结果，即将 data1 拷贝至 data3 处的结果

```
-g 002b
AX=0745 BX=0007 CX=0000 DX=0000 SP=0000 BP=0000 SI=0003 DI=0007
DS=076A ES=076B SS=0769 CS=076C IP=002B  NU UP EI PL NZ NA PO NC
076C:002B B024          MDU    AL,24
-D DS:0
076A:0000 43 44 45 41 42 43 44 00-00 00 00 00 00 00 00 00  CDEABCD.....
076A:0010 41 42 43 44 43 44 45 00-00 00 00 00 00 00 00 00  ABCDCDE.....
076A:0020 B8 6A 07 8E D8 B8 6B 07-8E C0 BB 03 00 BE 00 00  .j....k.....
076A:0030 BF 00 00 B9 04 00 8A 07-26 88 05 43 47 E2 F7 B9  .....&...CG...
076A:0040 03 00 8A 04 26 88 05 46-47 E2 F7 B0 24 26 88 05  ....&...FG...$&..
076A:0050 BC C3 8E DB 8D 16 00 00-B4 09 CD 21 B4 4C CD 21  .....!.L.!
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .@P.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P...s.....
```

由图可知，data3 所在段为 ES，偏移量为 0010，且通过循环将 data1 进行拷贝，从而存放数据为 CDE，对应 16 进制 ASCII 码值为 43 44 45，位置紧跟上一内容（ABCD）之后。

5.查看更换段基址指令的结果

```
-g 2f
AX=0745 BX=076B CX=0000 DX=0000 SP=0000 BP=0000 SI=0003 DI=0007
DS=076B ES=076B SS=0769 CS=076C IP=002F  NU UP EI PL NZ NA PO NC
076C:002F 8D160000      LEA    DX,[0000]          DS:0000=4241
-d ds:0
076B:0000 41 42 43 44 43 44 45 24-00 00 00 00 00 00 00 00  ABCDCDE$.
076B:0010 B8 6A 07 8E D8 B8 6B 07-8E C0 BE 00 00 BB 03 00  .j....k.....
076B:0020 BF 00 00 B9 04 00 8A 07-26 88 05 43 47 E2 F7 B9  .....&...CG...
076B:0030 03 00 8A 04 26 88 05 46-47 E2 F7 8C C3 8E DB 8D  ....&...FG.....
076B:0040 16 00 00 B4 09 CD 21 B4-4C CD 21 83 3E 13 02 18  .....!.L.!>...
076B:0050 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076B:0060 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t...v...
076B:0070 79 83 C4 02 2B C0 5E 8B-E5 5D C3 90 55 8B EC 83  y...+.^...l..U...
```

由图可知，data3 的地址为 076A:0010，根据公式计算得出  $076AH \times 10H + 0010H = 076B0H$ ，即 076B:0000，所以 ES 的段基址为 076B，且已经通过 BX 移至 DS 处，所以 DS 中的内容为 076B，由于偏移量为 0，所以在地址 076B:0000 处存放这 data3 的内容 ABCDCDE\$，\$ 为结束符，且以 16 进制 ASCII 值的形式进行存储，所以为 41 42 43 44 43 44 45 24。

实验名称	实验三：已知有符号数组 ARRAY 有 6 个元素。试编制统计此数组相邻两元素符号变化次数的程序		
实验性质 (必修、选修)	必修	实验类型（验证、设计、创新、综合）	设计
实验课时	8	实验日期	2023.10.13
实验仪器设备以及实验软硬件要求	电脑、DosBox、Notepad++		



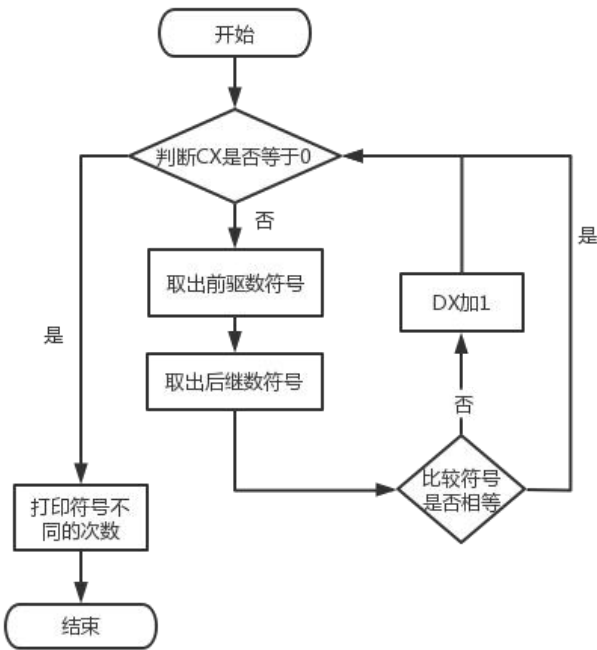
实验目的	<p>1.熟练掌握汇编语言的程序结构，学会如何使用数据定义、符号定义、模块定义等常用的批示性语句。</p> <p>2.熟悉汇编语言的工作环境；学会使用汇编语言的编辑、汇编、和连接装配程序的方法；熟悉汇编语言程序的编辑—汇编—连接—运行—调试的过程</p> <p>3.熟练掌握用 <b>DEBUG</b> 来调试程序，掌握调试程序的反汇编、读写寄存器或存储单元、运行程序等简单的调试命令。</p> <p>4.熟悉汇编语言指令，学会使用数据传送，串处理指令，处理器控制等简单指令。</p>
------	--

### 实验内容

#### 一、实验原理

通过比较前后两个数的符号变化进行计数，符号的变化用 **cmp** 指令进行判断，符号的类型根据最高位数字来进行判断，可以与 **8000H** 进行相与得出符号类型（**1** 代表负，**0** 代表正）。每比较一次根据数据定义类型进行后移相应的偏移量。

#### 二、程序流程图



#### 三、源程序（附注解）

```

1.  data segment
2.      array dw 0ffh,18h,8876h,5005,1008H,4321H
3.      n dw ?
4.  data ends
5.  code segment
6.      assume cs:code,ds:data
7.  main proc far

```

```

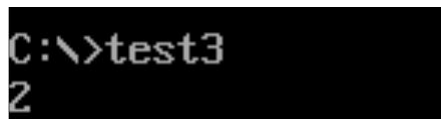
8.      start:
9.      push  ds
10.     sub   ax, ax
11.     push  ax
12.     mov   ax, data ;送段地址
13.     mov   ds, ax
14.
15.     mov   dx, 0 ;计数器
16.     mov   cx, 6 ;循环次数
17.     dec   cx
18.     lea   si, array ;获取 array 的偏移量
19. k:    mov   ax, [si] ;将该偏移量位置对应的数取出
20.     and   ax, 8000h ;判断该数的符号
21.     mov   bx, [si+2] ;将下一个数取出
22.     and   bx, 8000h ;也判断该数的符号
23.     cmp   ax, bx ;比较两数的符号
24.     je    l ;如果相等跳转至 l 直接进行下一次比较
25.     inc   dx ;如果不等说明符号有变化，进行计数
26. l:
27.     add   si, 2
28.     loop  k
29.     mov   n, dx
30.     mov   ah, 2
31.     add   dl, 30H
32.     int   21h
33.     ret
34. main  endp
35. code  ends
36. end   start

```

### 实验结果与分析

根据实验题目统计此数组相邻两元素符号变化次数，并根据实验要求进行调试，以下是过程截图以及结果说明。

运行结果：



```

C:\>test3
2

```

1.用 U 命令反汇编，了解指令地址

```

C:\>debug test3.exe
-u
076B:0000 1E          PUSH    DS
076B:0001 2BC0         SUB     AX,AX
076B:0003 50          PUSH    AX
076B:0004 B86A07       MOV     AX,076A
076B:0007 8ED8       MOV     DS,AX
076B:0009 BA0000       MOV     DX,0000
076B:000C B90600       MOV     CX,0006
076B:000F 49          DEC     CX
076B:0010 8D360000     LEA     SI,[0000]
076B:0014 8B04       MOV     AX,[SI]
076B:0016 250080       AND     AX,8000
076B:0019 8B5C02       MOV     BX,[SI+02]
076B:001C 81E30080     AND     BX,8000

```

## 2.查看数据初始状态

```

-g 000F
AX=076A BX=0000 CX=0006 DX=0000 SP=FFFC BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000F  NU UP EI PL ZR NA PE NC
076B:000F 49          DEC     CX
-d ds:0
076A:0000 FF 00 18 00 76 88 8D 13-08 10 21 43 00 00 00 00  ....v.....!C....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 BA 00 00 B9 06 00 49  .+.P.j.....I
076A:0020 8D 36 00 00 8B 04 25 00-80 8B 5C 02 81 E3 00 80  .6....%....\....
076A:0030 3B C3 74 01 42 83 C6 02-E2 EA 89 16 0C 00 CB 08  ;.t.B.....
076A:0040 B8 FF FF 5E 8B E5 5D C3-B8 80 01 50 FF 76 04 E8  ...^...l....P.v..
076A:0050 56 73 83 C4 04 89 46 FC-40 75 20 83 3E 13 02 18  U$....F.@u .>...
076A:0060 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t..v...

```

由图可知，由于 array 的数据类型为 DW，占位 2 个字节，数据有 0FFH，18H，8876H，5005，1008H，4321H。且 5005 为十进制数，转换为 16 进制数为 138D，所以在地址 076A:0000 存放的内容为 FF00 1800 7688 8D13 0810 2143，其中左侧字节为低字节，右侧为高字节。此时，CX 里存放的是循环次数 6。

## 3.查看数据结果状态

```

-g 002E
AX=0000 BX=0000 CX=0000 DX=0002 SP=FFFC BP=0000 SI=000A DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=002E  NU UP EI PL NZ NA PE NC
076B:002E CB          RETF
-d ds:0
076A:0000 FF 00 18 00 76 88 8D 13-08 10 21 43 02 00 00 00  ....v.....!C....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 BA 00 00 B9 06 00 49  .+.P.j.....I
076A:0020 8D 36 00 00 8B 04 25 00-80 8B 5C 02 81 E3 00 80  .6....%....\....
076A:0030 3B C3 74 01 42 83 C6 02-E2 EA 89 16 0C 00 CB 08  ;.t.B.....
076A:0040 B8 FF FF 5E 8B E5 5D C3-B8 80 01 50 FF 76 04 E8  ...^...l....P.v..
076A:0050 56 73 83 C4 04 89 46 FC-40 75 20 83 3E 13 02 18  U$....F.@u .>...
076A:0060 75 13 E8 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83  u...<...P.v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B  ...F...~...t..v...

```

由图可知，此时已经结束循环，并且统计了符号变化次数。n 与 array 定义在同一数据段中，所

以位置紧随其后，且变换次数为 2，所以为 0200，此时前一操作 MOV n, DX，使得 DX 也为 2。

实验名称	实验四：有一个首地址为 ARRAY 的 5 字数组，试编制一个程序，删除数组中所有为零的项，并将后续项向前压缩，最后将数组的剩余部分补上零。		
实验性质 (必修、选修)	必修	实验类型（验证、设计、创新、综合）	设计
实验课时	8	实验日期	2023.10.13
实验仪器设备以及实验软硬件要求	电脑、DosBox、Notepad++		
实验目的	<div>1.熟练掌握汇编语言的程序结构，学会如何使用数据定义、符号定义、模块定义等常用的批示性语句。</div> <div>2.熟悉汇编语言的工作环境；学会使用汇编语言的编辑、汇编、和连接装配程序的方法；熟悉汇编语言程序的编辑—汇编—连接—运行—调试的过程</div> <div>3.熟练掌握用 DEBUG 来调试程序，掌握调试程序的反汇编、读写寄存器或存储单元、运行程序等简单的调试命令。</div> <div>4.熟悉汇编语言指令，学会使用数据传送，串处理指令，处理器控制等简单指令。</div>		

实验内容

一、实验原理

数据段存数据的值和数组的长度，程序段依次比较数组每一位，当数组元素为 0 时，将后面的元素向前移动一位，并在末尾置 0，当数组元素不为 0 时，比较下一位，最后输出结果。

二、程序流程图

```
graph TD; Start([开始]) --> IsZero{元素是否为0}; IsZero -- 是 --> Shift[后继元素往前移一位，最后一个元素置为0]; IsZero -- 否 --> IsCXZero{判断CX是否为0}; Shift --> IsCXZero; IsCXZero -- 是 --> End([结束]); IsCXZero -- 否 --> CheckLast[校验最后元素]; CheckLast --> IsZero;
```

### 三、源程序（附注解）

```
1.  data_seg segment
2.      n dw 5
3.      ary dw 1,0,2,0,0
4.  data_seg ends
5.
6.  code_seg segment
7.  main proc far
8.      assume cs:code_seg,ds:data_seg
9.      start:
10.     push ds
11.     sub ax,ax
12.     push ax
13.     mov ax,data_seg
14.     mov ds,ax ;送段地址
15.     mov cx,n ;判断次数
16.     dec cx
17.     mov bx,0
18.     mov di,0 ;偏移地址放置在 di
19.     end1:
20.     cmp ary[di],0 ;将偏移地址对应的数取出与 0 进行比较
21.     je move ;如果相等跳转至 move 处
22.     add di,2
23.     dec cx
24.     jmp end1
25.     move:
26.     push cx ;压栈保留数据
27.     push di
28.     k:
29.     mov dx,ary[di+2]
30.     mov ary[di],dx ;将后一项往前移动一位
31.     add di,2
32.     loop k
33.     pop di
34.     pop cx
35.     inc bx
36.     loop end1
37.     mov cx,bx ;将 0 的个数放置在 CX
38.     end2:
39.     add di,2
40.     mov ary[di],0 ;在 ary 中添加 CX 数量的 0
41.     loop end2
42.     mov cx,n
43.     lea di,ary
```

```

44.  print:
45.  mov dx,[di]
46.  add dl,30h
47.  mov ah,2
48.  int 21h
49.  add di,2
50.  loop print    ;循环打印元素
51.  ret
52.  main endp
53.  code_seg ends
54.  end start

```

### 实验结果与分析

根据实验题目删除数组中所有为零的项，并将后续项向前压缩，最后将数组的剩余部分补上零，并根据实验要求进行调试，以下是过程截图以及结果说明。

运行结果：

```

C:\>test4
12000

```

1.用 U 命令反汇编，了解指令地址

```

C:\>debug test4.exe
-u
076B:0000 1E          PUSH     DS
076B:0001 2BC0          SUB      AX,AX
076B:0003 50          PUSH     AX
076B:0004 B86A07       MOV      AX,076A
076B:0007 8ED8       MOV      DS,AX
076B:0009 8B0E0000     MOV      CX,[0000]
076B:000D 49          DEC      CX
076B:000E BB0000     MOV      BX,0000
076B:0011 BF0000     MOV      DI,0000
076B:0014 83BD020000   CMP      WORD PTR [DI+0002],+00
076B:0019 7406       JZ       0021
076B:001B 83C702       ADD      DI,+02
076B:001E 49          DEC      CX
076B:001F EBF3       JMP      0014

```

2.查看数据初始状态

```

-g 000D
AX=076A BX=0000 CX=0005 DX=0000 SP=FFFC BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000D  NU UP EI PL ZR NA PE NC
076B:000D 49          DEC      CX
-d ds:0
076A:0000 05 00 01 00 00 00 02 00-00 00 00 00 00 00 00 .....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 8B 0E 00 00 49 BB 00 .+.P.j.....I..
076A:0020 00 BF 00 00 83 BD 02 00-00 74 06 83 C7 02 49 EB .....t....I.
076A:0030 F3 51 57 8B 95 04 00 89-95 02 00 83 C7 02 E2 F3 .QW.....
076A:0040 5F 59 43 E2 DF 8B CB 83-C7 02 C7 85 02 00 00 00 _YC.....
076A:0050 E2 F5 8B 0E 00 00 8D 3E-02 00 8B 15 80 C2 30 B4 .....>.....0.
076A:0060 02 CD 21 83 C7 02 E2 F2-CB FF 76 04 E8 39 73 83 ..!.....v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B ...F...~..t..v...

```

由图可知,由于数据段存放顺序是先 n 后 ary,且存放数据类型为 DW,所以存储内容为 0500 0100 0000 0200 0000 0000,且此时已经将 n 的内容移至 CX,所以 CX 为 0005。

### 3.查看数据结果状态

```

-g 0058
12000
AX=0230 BX=0002 CX=0000 DX=0030 SP=FFFC BP=0000 SI=0000 DI=000C
DS=076A ES=075A SS=0769 CS=076B IP=0058  NU UP EI PL NZ NA PE NC
076B:0058 CB          RETF
-d ds:0
076A:0000 05 00 01 00 02 00 00 00-00 00 00 00 00 00 00 .....
076A:0010 1E 2B C0 50 B8 6A 07 8E-D8 8B 0E 00 00 49 BB 00 .+.P.j.....I..
076A:0020 00 BF 00 00 83 BD 02 00-00 74 06 83 C7 02 49 EB .....t....I.
076A:0030 F3 51 57 8B 95 04 00 89-95 02 00 83 C7 02 E2 F3 .QW.....
076A:0040 5F 59 43 E2 DF 8B CB 83-C7 02 C7 85 02 00 00 00 _YC.....
076A:0050 E2 F5 8B 0E 00 00 8D 3E-02 00 8B 15 80 C2 30 B4 .....>.....0.
076A:0060 02 CD 21 83 C7 02 E2 F2-CB FF 76 04 E8 39 73 83 ..!.....v..9s.
076A:0070 C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B ...F...~..t..v...

```

由图可知,通过移位得到数组的结果为 12000,且存放数据类型为 DW,所以存储内容为 0500 0100 0200 0000 0000 0000。