

热山大学

汇编语言程序设计实验指导书

Assembler language Programming Experiment Instruction Book

> 教 务 处 2024 年 9 月

实验一 汇编语言源程序的输入

一、实验目的

- 1. 通过实验了解和熟悉微机系统的配置。
- 2. 学习在 DEBUG 状态下输入汇编源程序的方法。
- 3. 初步掌握调试(在 DEBUG 状态下)的过程。

二、实验原理

- 1. 本实验要求在 DEBUG 状态下输入汇编源程序,并用 DEBUG 命令进行调试。用单步跟踪的方法验证指令的功能。
 - 2. 以下是给定的参考程序,在实验时需按要求填写相应指令的执行结果。
- 注:(1) 微机进入 DEBUG 状态下之后,一切立即数和地址数据均被默认为十六进制数,在输入时数的后面不加后缀"H":
- (2) 在 DEBUG 状态下执行程序时,"INT 20H"指令可使系统执行完该指令前的程序时返回到"-"提示符状态,并且恢复 CS 和 IP 寄存器原来的值。

三、实验仪器

微机一台。

四、实验步骤

- 1. 开机后进入 DOS 系统,
 - C> DEBUG ∠ (∠回车符)
 - (为 **DEBUG** 提示符)

当显示器出现提示符"一"时,说明已进入 DEBUG 状态,这时,可用 DEBUG 命令进行操作。

2. 用 DEBUG 的 Register 命令检查所有寄存器内容,并作记录。命令格式:

R [寄存器名]

该命令的功能是显示寄存器的内容,或修改某一指定寄存器内容,若[寄存器名]缺省,则显示所有寄存器内容。例如:

-R

3. 用 DEBUG 的 Assemble 命令输入汇编源程序。格式:

A[内存地址]

注:用"[]"符号括起来的部分表示可以省略。

该命令的功能是从指定的内存地址开始(方括号不要输入)逐条输入汇编语言源程序, 并汇编成机器码存入内存。若地址缺省,则接上一个A命令最后一条指令之后输入汇编语句,若没有用过A命令,则从CS: 0100H地址开始输入。例如:

-A

0CD3: 0100-

在输入 A 命令之后,或每输入一条指令之后,显示器的左端给出了内存的段地址和偏移地址。

每条指令均用回车(∠)结束。若输入的指令有语法错误,DEBUG 拒绝接收,并给出提示,此时可以重新输入。程序的最后一条指令输入完之后,再按一次回车键(∠),即可结束汇编命令,回到 DEBUG 提示符"一"状态。

4. 用 DEBUG 的 Unassemble 命令反汇编。命令格式:

U[起始地址[终止地址]]

该命令的功能是从起始地址到终止地址反汇编目标码,缺省值是接上一个 U 命令或从 CS: 0100H 地址开始。例如:

-U

显示器上将显示程序的内存地址、指令机器码的汇编源程序三列对照清单。

5. 用 DEBUG 的 Trace 命令单步跟踪程序。命令格式:

T[=起始地址][指令条数]

该命令的功能是从指定的起始地址开始逐条执行指令,每执行完一条指令,屏幕显示 所有寄存器内容和下一条指令地址和指令。若[=起始地址]缺省,则 T 命令从 CS: IP 地址 开始执行指令。例如:

 $-T \angle$

重复这一过程,即可看到每条指令执行后所有寄存器和标志寄存器的标志位内容。此时,要检查内存单元的数据,可用 DEBUG 的 D 命令。

6. 用 DEBUG 的 Dump 命令显示存贮器单元的内容。命令格式:

D[起始地址[终止地址]]

该命令的功能是从起始地址到终止地址,连续显示存贮器单元的内容。若地址缺省,则接上一个 D 命令或从 DS: 0100H 地址开始显示。例如:

 $-D \checkmark$

参考程序:

MOV AX, 2000 AL= MOV DS, AX DS= NOT AX AX= XOR AX, AX AX= DEC AX AX= INC AX AX= MOV BX, 1040 BH= MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BL, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX	<i>□</i> 3/1±/1•	
NOT AX AX= XOR AX, AX AX= DEC AX AX= INC AX AX= MOV BX, 1040 BH= MOV SI, BX SI= MOV [1040H]= [1041H]= MOV WORD PTR[SI], 20F [1040H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BL, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV AX, 2000	AL=
XOR AX, AX AX= DEC AX AX= INC AX AX= MOV BX, 1040 BH= MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= PUSH AX AX=	MOV DS, AX	DS=
DEC AX AX= INC AX AX= MOV BX, 1040 BH= MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= PUSH AX AX=	NOT AX	AX=
INC AX AX= MOV BX, 1040 BH= MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= [1041H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= PUSH AX AX=	XOR AX, AX	AX=
MOV BX, 1040 BH= MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	DEC AX	AX=
MOV SI, BX SI= MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= [SS: 1FFEH]= PUSH AX AX= [SS: 1FFEH]=	INC AX	AX=
MOV [SI], BL [1040H]= MOV WORD PTR[SI], 20F [1040H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV BX, 1040	BH=
MOV WORD PTR[SI], 20F [1040H]= [1041H]= MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV SI, BX	SI=
MOV DI, SI DI= MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV [SI], BL	[1040H]=
MOV [DI+50], BH [DI+50H]= MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV WORD PTR[SI], 20F	[1040H]= [1041H]=
MOV BH, [SI] BH= MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV DI, SI	DI=
MOV BL, [DI+50] BL= MOV SP, 2000 PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV [DI+50], BH	[DI+50H]=
MOV SP, 2000 PUSH AX	MOV BH, [SI]	BH=
PUSH AX AX= [SS: 1FFEH]= [SS: 1FFFH]=	MOV BL, [DI+50]	BL=
	MOV SP, 2000	
	PUSH AX	AX= [SS: 1FFEH]= [SS: 1FFFH]=
PUSH BX BX= [SS: 1FFCH]= [SS: 1FFDH]=	PUSH BX	BX= [SS: 1FFCH]= [SS: 1FFDH]=
POP AX AX=	POP AX	AX=
POPF F=	POPF	F=
NEG BX BX=	NEG BX	BX=
XCHG BX, AX BX=	XCHG BX, AX	BX=
STD F=	STD	F=
STI F=	STI	F=
CLD F=	CLD	F=
CLI F=	CLI	F=
ADC DI, 2050 DI= F=	ADC DI, 2050	DI= F=
ADC SP, DI SP= F=	ADC SP, DI	SP= F=

ADC	AX, 1500	AX=	F=
SUB	AX, BX	AX=	BX=
SHL	AH, 1	AH=	
RCL	AX, 1	AX=	
SHR	BH, 1	BH=	
RCR	BL, 1	BL=	
MOV	CL, 8		
MOV	DX, 80F0		
ROL	DX, CL	DX=	CL=

注:上表中 F 的取值要按照下表中的对应关系转换成 8 位二进制数后再填到相应位置

标志名	标志为1	标志为 0
OF 溢出 (是/否)	O V	NV
DF 方向 (减量/增量)	DN	UP
IF 中断 (允许/关闭)	ΕI	DI
SF 符号 (负/正)	NG	PL
ZF 零 (是/否)	ZR	NZ
AF辅助进位(是/否)	A C	NA
PF 奇偶 (偶/奇)	PE	PO
CF 进位(是/否)	CY	NC

五、实验要求

- 1. 整理每条指令执行的结果,填到打印清单的右半部分(应注意内存数据检查的正确性)。
 - 2. 比较实验记录与理论分析的结果是否相同,若有不同,找出差别及问题所在。
 - 3. 总结本次实验的体会。

实验二 数据的建立与传送程序

一、实验目的

- 1. 继续学习 DEBUG 命令。
- 2. 验证指令的功能。

二、实验原理

在 DEBUG 状态下,分别输入下面各程序段,每输入完一个程序段,用 G 命令进行连续方式执行程序,在连续执行时,要记录程序的执行结果。

参考程序:

1. 在内存 10000H 单元开始,建立 $00H\sim0FH\sim00H$ 31 个数,要求 $00H\sim0FH$ 数据逐渐增大, $0FH\sim00H$ 逐渐减小。该程序从内存 CS:0100H 地址开始输入。

MOV AX, 1000H

MOV DS, AX

MOV SI, 0

MOV CL, 0FH

XOR AX, AX

PPE1: MOV [SI], AL

INC SI

INC AL

DEC CL

JNZ PPE1

MOV CX, 10H

PPE2: MOV [SI], AL

INC SI

DEC AL

LOOP PPE2

INT 20H

注:转移指令的符号地址直接用绝对偏移地址,该地址在用 A 命令汇编输入时,可以看到程序全部运行完之后,可用 DEBUG 的 Dump 命令查看建立的数据块内容。例如:

-D1000:00 1E

- 2. 把上一个程序的执行结果(建立的 31 个字节数据块, 其首地址在 10000H), 分几种方式传送到以下指定的区域。
- (a) 该程序从内存 CS:0200H 开始输入。把数据块传送到 **1000:2050**(即 12050H)开始的存储区域中。

参考程序: MOV AX, 1000H

MOV DS. AX

MOV SI, 0

MOV DI, 2050H

MOV CX, 1FH ; 数据块长度是 31

PPEA: MOV AL, [SI]

MOV [DI], AL

INC SI

INC DI

LOOP PPEA

INT 20H

检查内存数据块的传送情况,可用"D"命令。

(b) 用串传送指令 MOVSB, 把数据块传送到 **1500:学号后 4** 位开始的区域,该程序从内存 CS:0250H 开始输入。(例如学号后 4 位为 0001,那么就把数据块传送到 **1500:0001**即 **15001H** 开始的存储区域中)

检查程序最后的执行结果,可用"D"命令,例如:

- -D1500:学号后 4 位 ✓
- (c) 用重复串操作指令"REP MOVSB"把数据块传送到 2000:学号后 4 位开始的区域。 该程序从 CS:0300H 地址开始输入。

检查程序的最后执行结果时,可用:

- -D2000: 学号后 4 位∠
- (d) 用串操作的减量工作方式,把数据块传送到 **2500:学号后 4 位**开始的区域。该程序 从 CS:0350H 开始输入。

检查程序的最后执行结果,用D命令:

-D2500:学号后 4 位∠

三、实验仪器

微机一台。

四、实验步骤

参照实验一的步骤,按照本实验程序的内容,分别输入各段程序,连续执行程序,记录每个程序段的最后结果。若想把源数据块重新换一批数据,可以用 DEBUG 的 Fill 命令填充新的数据。例如:

-F1000:00 L1F 33 ✓

从 1000:0000H 开始的 31 个字节被替换成 33H。

五、实验预习要求

- 1. 认真阅读 DEBUG 调试程序的命令说明。
- 2. 仔细阅读本实验给出的程序段。
- 3. 写出每个程序段的结果数据(分析结果),并说明结果所在的位置(寄存器或存储器)。

六、实验报告要求

- 1. 整理每个程序段在实验时的记录内容。<mark>每个程序段和运行结果都以截图的形式插入</mark>相应的位置。
 - 2. 比较每个程序段的特点。
 - 3. 分析本次实验出现的问题, 找出问题所在。

实验三 分支程序设计

一、实验目的

- 1. 练习分支程序的编写方法。
- 2. 练习汇编语言程序的上机过程。

二、实验原理

- 1. 通过分支程序设计调试和运行,进一步熟悉掌握汇编程序执行的软件环境。
- 2. 通过分支程序的执行过程,熟悉 EDIT 的使用,建立 OBJ 文件 EXE 文件的方法。

三、实验仪器

微机一台。

四、实验内容

1、首先测试字节变量 VAR 的最高位,如果为 1,则显示字母"L";如果最高位不为 1,则继续测试最低位,如果最低位为 1,则显示字母"R";如果最低位也不为 1,则显示字母"M"。

注:程序中 VAR 变量的值预设为自己学号的后两位(例如,自己学号后两位为 17,那么 VAR 的值就是 17H)

2、首先从键盘上输入一个字符串到输入缓冲区,然后再任意输入一个字符,编程记录该字符在输入字符串中出现的次数,将结果保存在变量 num 中,要求在屏幕输出显示输入的字符串、字符及变量 num 的值。

五、实验要求

实验报告中要有源程序和执行结果截图。

实验四 统计学生成绩程序

一、实验目的

进一步掌握分支程序和循环程序的编写方法。

二、实验原理

设有 10 个学生的成绩分别为 70、60、89、76、69、75、99、59、100 和 80 分。试编制程序分别统计低于 60 分、 $60\sim69$ 分、 $70\sim79$ 分、 $80\sim89$ 分、 $90\sim99$ 分及 100 分的人数存放到 s5、s6、s7、s8、s9 及 s10 单元中。

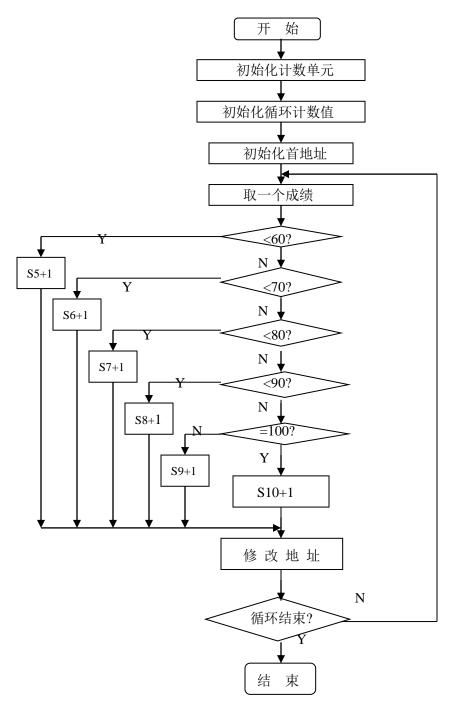
这一题目的算法很简单,成绩分等部分采用分支结构,统计所有成绩则用循环结构完 成。程序框图如下图所示。

三、实验仪器

微机一台。

四、实验步骤

参考给的流程图编写源程序,进行调试。 流程图如下:



程序框图

参考程序如下:

; PROGRAM TITLE GOES HERE——result

ale ale ale ale	ale ale ale ale ale a	de ale ale ale ale ale ale ale ale ale al	ate ale ale ale ale ale ale ale ale ale al	
; ************************************			; DEFINE DATA SEGMENT	
GRAD			9,75,99,59,100,80	
			,,13,,27,,32,,100,00	
	DW	0		
~ ~	DW	0		
	DW	0		
S9		0		
	DW	0		
~	A ENDS	U		
		* * * * * * * * * * * *	*****	
,				
PROGNA	M SEGME		; DEFINE CODE SEGMENT	
;	DD OC	EAD		
	PROC		; MAIN PART OF PROGRAM	
	ASSUME CS:PROGNAM, DS:DATAREA			
~	START: ; STARTING EXECUTION ADDRESS			
; SET UP STACK FOR RETURN				
	PUSH	_ 2	; SAVE OLD DATA SEGMENT	
	SUB		; PUT ZERO IN AX	
PUSH AX ; SAVE IT ON STACK			; SAVE IT ON STACK	
; SET DS REGISTER TO CURRENT DATA SEGMENT				
	MOV	AX,DATAREA	; DATAREA SEGMENT ADDR	
	MOV	DS,AX	; INTO DS REGISTER	
; MAIN PART OF PROGRAM GOES HERE				
	MOV	S5, 0	; INITIALIZE COUNTER	
	MOV	S6, 0		
	MOV	S7, 0		
	MOV	S8, 0		
	MOV	S9, 0		

MOV

MOV

MOV

S10, 0

CX, 10

; INITIALIZE LOOP COUNT VALUE

BX, OFFSET GRADE ; INITIALIZE FIRST ADDR

COMPARE:

MOV AX,[BX] ; GET A RESULT

CMP AX, 60; <60?

JL FIVE

CMP AX, 70 ; <70?

JL SIX

CMP AX, 80 ; <80?

JL SEVEN

CMP AX, 90 ; <90?

JL EIGHT

CMP AX, 100 ; =100?

JNE NINE

INC S10

JMP SHORT CHANGE_ADDR

NINE: INC S9

JMP SHORT CHANGE_ADDR

EIGHT: INC S8

JMP SHORT CHANGE_ADDR

SEVEN: INC S7

JMP SHORT CHANGE_ADDR

SIX: INC S6

JMP SHORT CHANGE_ADDR

FIVE: INC S5

CHANGE_ADDR:

ADD BX, 2

LOOP COMPARE

RET ; RETURN TO DOS

MAIN ENDP ; END OF MAIN PART OF PROGRAM

: -----

PROGNAM ENDS

END START ; END ASSEMBLY

五、实验要求

- 1. 读懂所给的程序。
- 2. 编写将存放于 S5、S6、S7、S8、S9 及 S10 单元中的数据在屏幕上显示出来的部分程序,并记录运行结果。该部分程序要求利用循环结构实现,并应明确指出将该段程序插入哪个位置。

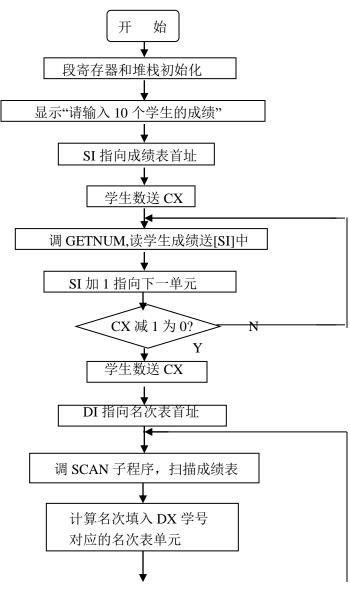
实验五 学生成绩名次表实验

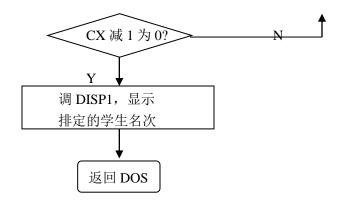
一、实验目的

进一步熟悉排序方法。

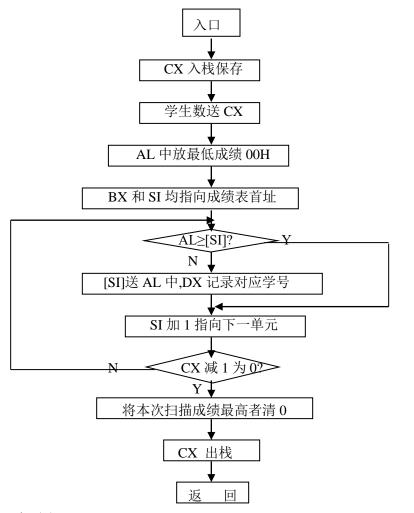
二、实验原理

将 0~100 之间的 10 个成绩存入首址为 1000H 的单元中。1000H+i 表示学号为 i 的学生成绩,编写程序能在 2000H 开始的区域排出名次表。2000H+i 为学号 i 的学生的名次。参考主程序:





程序 SCAN:



三、实验仪器

微机一台。

四、实验步骤

- 1. 参考流程图编写实验程序,进行调试。
- 2. 记录实验结果。

五、实验要求

参考程序中输入的成绩和显示的排名都是十六进制数,要求读懂程序后将其都修改为 十进制数,或者自己重新编写一个学生成绩排名程序。

附参考程序:

CRLF	MACRO
MOV	AH, 02H
MOV	DL, 0DH
INT	21H
MOV	AH, 02H
MOV	DL, 0AH
INT	21H

ENDM

DATA	SEGMEN	T
STUNUM	EQU	10
MESS	DB	'INPUT 10 STUDENTS SCORE: ',0DH,0AH,'\$'

ERROR DB 'INPUT ERROR!',0DH,0AH,'\$'

ORG 1000H

SCORE DB 10 DUP (?)

ORG 2000H

SEQU DB 10 DUP (?)

DATA ENDS

STACK SEGMENT

STA DW 12 DUP (?)

TOP DW ?

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,ES:DATA,SS:STACK

START: MOV AX, DATA

MOV DS, AX

MOV ES, AX

MOV SP, TOP ; 初始化

MOV AH, 09H

MOV DX, OFFSET MESS

INT 21H ; 显示提示信息

MOV SI, OFFSET SCORE ; 成绩表首址

MOV CX, STUNUM ; 学生数送 CX

UUU: CALL GETNUM ; 读取键入数值送 DX

MOV [SI], DL ; 存入成绩表缓冲区

INC SI ;指向下一单元

LOOP UUU

MOV CX, STUNUM ; 学生数

MOV DI, OFFSET SEQU ; 名次表首址

VVV: CALL SCAN : 扫描子程序

MOV AL, STUNUM ; 学生数

SUB AL, CL

INC AL ; 计算名次

MOV BX, DX

MOV [DI+BX], AL ; 记 DX 学号对应名次

LOOP VVV

MOV CX, STUNUM ; 学生数

MOV SI, OFFSET SEQU ; 名次表首址

WWW: MOV AL, [SI]

CALL DISP1

CRLF

INC SI

LOOP WWW ;显示排定的学生名次

MOV AX, 4C00H

INT 21H

SCAN PROC NEAR ; 子程序,每扫描一遍成绩表缓冲区, 找出

其中成绩最高者由 DX 指针指示对应学生

之后将该成绩清除,以便下一次扫描

PUSH CX

MOV CX, STUNUM ; 学生数

MOV AL, 00H ; 最低成绩

MOV BX, OFFSET SCORE

MOV SI, BX ; 指向成绩表首址

CCC: CMP AL, [SI]

JAE JJJ ; AL中的成绩不低于成绩表指针SI

所指单元的成绩则转JJJ

MOV AL, [SI] ; AL 存放较高的成绩

MOV DX, SI

SUB DX, BX ; DX 为对应学号

JJJ: INC SI ; 指向下一单元

LOOP CCC

ADD BX, DX

MOV BYTE PTR[BX], 00H ; 本次扫描成绩最高者清 0

POP CX

RET

SCAN ENDP

DISP1 PROC NEAR

PUSH CX

MOV BL, AL

MOV DL, BL

MOV CL, 04

ROL	DL, CL
AND	DL, 0FH
CALL	DISPL
MOV	DL, BL
AND	DL, 0FH
CALL	DISPL
POP	CX
RET	
ENDP	
PROC	NEAR
ADD	DL, 30H

CMP DL, 3AH

JB DDD

ADD DL, 27H

MOV AH, 02H

INT 21H

RET

DISPL ENDP

DISP1

DISPL

DDD:

GETNUM PROC NEAR

PUSH CX

XOR DX, DX

GGG: MOV AH, 01H

INT 21H

CMP AL, 0DH

JZ PPP

CMP AL, 20H

JZ PPP

SUB AL, 30H

JB KKK

CMP AL, 0AH

JB GETS

CMP AL, 11H

JB KKK

SUB AL, 07H

CMP AL, 0FH

JBE GETS

CMP AL, 2AH

JB KKK

CMP AL, 2FH

JA KKK

SUB AL, 20H

GETS: MOV CL, 04

SHL DX, CL

XOR AH, AH

ADD DX, AX

JMP GGG

KKK: MOV AH, 09H

MOV DX, OFFSET ERROR

INT 21H

PPP: PUSH DX

CRLF

POP DX

POP CX

RET

GETNUM ENDP

CODE ENDS

END START

封面设计: 贾丽

地 址:中国河北省秦皇岛市河北大街 438 号

邮 编: 066004

电话: 0335-8057068 传真: 0335-8057068

网址: http://jwc.ysu.edu.cn