

全国计算机等级考试

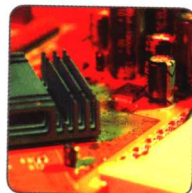
2005

历届上机真题详解

三级汇编语言程序设计



全国计算机等级考试命题研究组 编



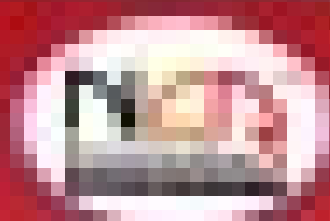
南开大学出版社

全国计算机等级考试

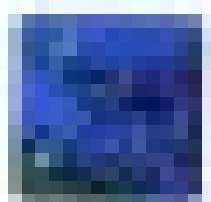
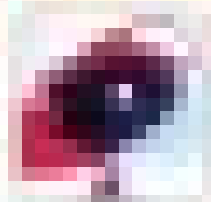
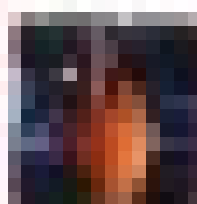
2005

历届上机真题详解

三级汇编语言程序设计



全国计算机等级考试教材系列



编 著 李天祥等

全国计算机等级考试

历届上机真题详解

三级汇编语言程序设计

全国计算机等级考试命题研究组 编

南开大学出版社

天 津

内容提要

本书提供了全国计算机等级考试三级汇编语言机试试题库, 共计有 100 道汇编考试题。书中对这 100 道题提供了答案, 并进行详细、准确的分析, 指出考核的知识点、重点、难点、解题思路、程序流程。所有题目均进行了上机测试, 保证能够在实际环境中正常运行。

本书配套光盘中有汇编语言考试的全真模拟机试环境, 并配有大量全真模拟试题, 可用于考前实践训练。另外, 光盘中还提供了 100 套试题的源代码, 便于读者练习。

本书针对参加全国计算机等级考试三级 PC 技术的考生, 同时也可以作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

图书在版编目(CIP)数据

全国计算机等级考试历届上机真题详解. 三级汇编语言程序设计 / 全国计算机等级考试命题研究组 编. 天津: 南开大学出版社, 2005. 4

ISBN 7-310-02286-6

I. 全... II. 全... III. ①电子计算机—水平考试—解题②汇编语言—程序设计—水平考试—解题
IV. TP3—44

中国版本图书馆 CIP 数据核字(2005)第 004115 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

营销部传真: (022)23508542 邮购部电话: (022)23502200

*

天津市蓟县宏图印务有限公司印刷

全国各地新华书店经销

*

2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

787×1092 毫米 16 开本 18.5 印张 442 千字

定价: 34.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022)23507125

前 言

全国计算机等级考试（National Computer Rank Examination, NCRE）是由教育部考试中心主办，用于考查应试人员的计算机应用知识与能力的考试。本考试的证书已经成为许多单位招聘员工的一个必要条件，具有相当的“含金量”。

为了帮助考生更顺利地通过计算机等级考试，我们作了大量市场调查，根据考生的备考体会，以及培训教师的授课经验，推出了《历届上机真题详解——三级汇编语言程序设计》。本书主要有三个组成部分。

一、100 套汇编语言题

对于备战等级考试而言，做真题，是进行考前冲刺的最佳方式。这是因为它的针对性相当强，考生可以通过真题的实际练习，来检验自己是否真正掌握了相关知识点，了解了考试重点，并且根据需要再对知识结构的薄弱环节进行强化。

二、详解真题

在每套题的后面，都有针对各个试题的答案和详细分析。在解析中，精解考点，分析题眼，详解重点难点，并给出应试技巧。

三、笔试和机试全真模拟环境配套光盘

第三部分是本书配套光盘，其中有本考试的全真模拟笔试和上机环境，并配有大量全真模拟试题，可用于考前实战训练。另外，光盘中还提供了 100 套试题的源代码，便于读者练习。

本书针对参加全国计算机等级考试三级 PC 技术的考生，同时也可以作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

为了保证本书及时面世和内容准确，很多朋友做出了贡献，陈河南、陈占军、许伟、赵晓睿、裘蕾、薛飞、李齐鹏、贺军、侯佳宜、戴文雅、戴军、贺民、何雄、周虎、刘大为、夏绪虎、邵世磊、任世华、汤效平、冯苗、王淼、刘占坤、李东锋、张新苗、李季、邹晓东、于樊鹏、梁彩隆、赵世伟、郑炎、黄志雄、邱代燕、宫晓琳、田仙仙、王莹、李志云、陈安南、李晓春、王春桥、王雷、韦笑、龚亚萍等老师在编写文档、调试程序、排版、查错、预读、光盘制作与测试等工作中加班加点，付出了很多辛苦，在此一并表示感谢！

本书中的所有题目均进行了上机测试，保证能够在实际环境中正常运行。在学习的过程中，您如有意见或建议，请与我们联系：xiaoxiang-007@sohu.com。

全国计算机等级考试命题研究组

2004 年 12 月

★★

第 1 题

请编制程序，其功能是：将内存中由 SOURCE 指示的 40 个字节有符号数组成的数组分成正数和负数两个数组，并求这两个数组的数据个数，结果存放在 RESULT 指示的内存区域。存放形式为正数个数在前，其后跟正数数组元素，然后是负数个数及负数数组元素。

例如：

内存中有 1EH, 91H, 74H, 91H, 42H, 30H, 81H, F3H, 18H, 25H

结果为 06H, 1EH, 74H, 42H, 30H, 18H, 25H, 04H, 91H, 91H, 81H, F3H

部分程序已经给出，其中原始数据由过程 LOAD 从文件 INPUT1.DAT 中读入 SOURCE 开始的内存单元中，转换结果要求从 RESULT 开始存放，由过程 SAVE 保存到文件 OUTPUT1.DAT 中。

请填空 BEGIN 和 END 之间已经给出的一段源程序使其完整，需填空处已经用横线标出，每个空白一般只需要填一条指令或指令的一部分（指令助记符或操作数），也可以填入功能相当的多条指令，或删除 BEGIN 和 END 之间原有的代码并自行编程来完成所要求的功能。

对程序必须进行汇编，并与 IO.OBJ 链接产生可执行文件，最终运行程序产生结果。调试中若发现整个程序中存在错误之处，请加以修改。

试题程序：

```

                EXTRN    LOAD:FAR,SAVE:FAR
N               EQU      40

STAC            SEGMENT STACK
                DB       128 DUP (?)
STAC            ENDS

DATA            SEGMENT
SOURCE          DB       N DUP(0)
RESULT          DB       N+2 DUP(0)
NAME0           DB       'INPUT1.DAT',0
NAME1           DB       'OUTPUT1.DAT',0
NDATA          DB       N NUP(0)
PDATA          DB       N NUP(0)
DATA            ENDS

CODE            SEGMENT
                ASSUME    CS:CODE, DS:DATA, SS:STAC
START           PROC     FAR

```

	PUSH	DS	
	XOR	AX,AX	
	PUSH	AX	
	MOV	AX,DATA	
	MOV	DS,AX	
	MOV	ES,AX	;置附加段寄存器
	LEA	DX,SOURCE	;数据区起始地址
	LEA	SI,NAME0	;原始数据文件名
	MOV	CX,N	;字节数
	CALL	LOAD	;从'INPUT1.DAT'中读取数据
;	**** BEGIN ****		
	LEA	SI, RESULT	
	<u>(1)</u>	DI,OFFSET PDATA	;PDATA 为正数数组存放缓冲区首址
	MOV	BX,OFFSET NDATA	;NDATA 为负数数组存放缓冲区首址
	XOR	DX,DX	
	MOV	CX,N	
	CLD		
MAIN1:	LODSB		
	TEST	AL, <u>(2)</u>	
	JZ	MAIN2	
	INC	DH	
	MOV	[BX],AL	
	INC	BX	
		<u>(3)</u>	
MAIN2:	INC	DL	
	MOV	[DI],AL	
	INC	DI	
MAIN3:	<u>(4)</u>	MAIN1	
	LEA	SI,PDATA	
	LEA	DI,RESULT	
	MOV	[DI],DL	
	INC	DI	
	XOR	CX,CX	
	MOV	CL,DL	
MAIN4:	MOV	AL, <u>(5)</u>	
	MOV	[DI],AL	
	INC	DI	
	INC	SI	

```

                                LOOP      (6)
                                MOV        [DI],DH
                                INC        DI
                                XOR        CX,CX
                                MOV        CL,DH
                                MOV        BX,OFFSET NDATA
MAIN5:  MOV        AL,[BX]
                                MOV        [DI],AL
                                INC        DI
                                (7)
                                LOOP      MAIN5
;  **** END ****
                                LEA        DX,RESULT      ;结果数据区首址
                                LEA        SI,NAME1        ;结果文件名起始地址
                                MOV        CX,N+2          ;字节数
                                CALL        SAVE            ;保存结果到'OUTPUT1.DAT'文件
                                RET
START  ENDP
CODE   ENDS
                                END        START

```

- 【答案】(1) MOV
 (2) 80H
 (3) JMP MAIN3
 (4) LOOP
 (5) [SI]
 (6) MAIN4
 (7) INC BX

【分析】

此程序主要实现正数和负数的分类，以及统计正数和负数的个数。正数和负数的所有结果存放在寄存器 DI 中。寄存器 DL 存放正数的个数，DH 存放负数的个数。

程序的数据段中定义了六个字节型变量，原始数据区 SOURCE、结果数据区 RESULT、原始数据文件名 NAME0、结果数据文件名 NAME1，以及正数和负数数组存放缓冲区首址。数据段下面是代码段，告诉汇编程序，代码段、数据段和堆栈段分别属于段寄存器 CS、DS、SS。

在程序的开始部分，是对返回地址的保存和数据段寄存器的初始化，这是每一个汇编程序所必须具有的操作步骤。准备工作完成以后，程序进入主体部分。

程序的开始，将原始数据偏移地址装入到 SI 中。正数和负数存放缓冲区首址分别送 DI 和 BX 中，所以第 (1) 空填写“MOV”。然后用 LODSB 指令将 SI 当前所指向的地址

单元（即[SI]）中内容装入到 AL 中。接着要判断 AL 的正负性，第（2）空填写“80H”，将 AL 和 80H 做 TEST 指令操作。若 TEST 指令的结果 ZF=0，则存放负数个数的 DH 加 1，然后转向 MAIN3 段，所以第（3）空填写 JMP MAIN3；若 ZF=1，则 AL 为正数，转向 MAIN2 段，存放正数个数的 DL 加 1。

MAIN3 段开始处要填写第（4）空，显然应该是“LOOP MAIN1”，即继续判断下一个数 AL 的正负性。然后将正数个数 DL 送 CL，用于控制循环次数。再依次将 DL 个正数存入到 DI 当前所指向的地址单元（即[DI]）中。第（5）空填写“SI”，SI 装入的是正数的偏移地址。第（6）空考查的是循环，装入正数到[DI]的循环入口处是 MAIN4。

MAIN5 段是依次将负数存放 to [DI] 中，将负数个数 DH 赋值给 CL，控制装入负数的循环次数。因为 BX 中装入的是负数的偏移地址，所以每次存入一个负数后，偏移地址 BX 加 1，指向下一个负数。所以第（7）空填写“INC BX”。

END 后面的指令是分别将结果数据区首址、结果文件名装入到 DX 和 SI 中，并保存结果文件。

★★

第 2 题

请编制程序，其功能是：内存中连续存放着 10 个无符号 8 位格雷码表示法的数，现将此十个数转换成十个 8 位二进制数，结果存入内存。其转换方法为二进制数的最高位 d7 与格雷码的最高位 g7 相同，二进制数的其余七位 d_k ($k=6, \dots, 0$) 分别为格雷的位 g_k ($k=6, \dots, 0$) 与二进制数的位 d_{k+1} ($k=6, \dots, 0$) 异或的结果。

例如：

内存中有 00H, 03H, 2BH, 67H, 0CH, 15H, 54H, 02H, D8H, C7H

结果为 00H, 02H, 32H, 45H, 08H, 19H, 67H, 03H, 90H, 85H

部分程序已给出，其中原始数据由过程 LOAD 从文件 INPUT1.DAT 中读入 SOURCE 开始的内存单元中。运算结果要求从 RESULT 开始存放，由过程 SAVE 保存到文件 OUTPUT1.DAT 中。

请填空 BEGIN 和 END 之间已给出的源程序使其完整（空白已用横线标出，每个空白一般只需一条指令，但采用功能相当的多条指令亦可），或删除 BEGIN 和 END 之间原有的代码并自行编程来完成所要求的功能。

对程序必须进行汇编，并与 IO.OBJ 链接产生可执行文件，最终运行程序产生结果。调试中若发现整个程序中存在错误之处，请加以修改。

试题程序：

```

                EXTRN    LOAD:FAR,SAVE:FAR
N               EQU      10

                SEGMENT  STACK
                DB       128 DUP (?)

```

```

STAC      ENDS

DATA      SEGMENT
SOURCE    DB      N DUP(?)           ;顺序存放 10 个字节数
RESULT    DB      N DUP(0)           ;存放结果
NAME0     DB      'INPUT1.DAT',0
NAME1     DB      'OUTPUT1.DAT',0
DATA      ENDS

CODE      SEGMENT
          ASSUME    CS:CODE, DS:DATA, SS:STAC
START     PROC      FAR
          PUSH      DS
          XOR       AX,AX
          PUSH      AX
          MOV       AX,DATA
          MOV       DS,AX

          LEA       SI,SOURCE          ;数据区起始地址
          LEA       SI,NAME0           ;原始数据文件名
          MOV       CX,N               ;字节数
          CALL      LOAD               ;从'INPUT1.DAT'中读取数据
;      **** BEGIN ****
          LEA       DI, RESULT
          LEA       SI, SOURCE
          MOV       CX,10
AGN0:     MOV       AL,[SI]
          _____
          (1)
          MOV       CX,8
          MOV       BX,0
AGN1:     MOV       AH,0
          SHL       BL,1
          _____
          (2)
          AL,1
          RCL       AH,L
          CMP       AH,_____
          _____
          (3)
          _____
          (4)
          JMP       NEXT
SET_ONE:  OR        BL,01H
NEXT:     MOV       _____,BL
          _____
          (5)
          _____
          (6)

```

```

                LOOP    AGN1
                (7)
                MOV     [DI],BL
                INC     SI
                INC     DI
                LOOP    AGN0
;    **** END ****
                LEA     DX,RESULT    ;结果数据区首址
                LEA     SI,NAME1     ;结果文件名
                MOV     CX,N         ;结果字节数
                CALL    SAVE         ;保存结果到文件
                RET
START    ENDP
CODE     ENDS
                END      START

```

- 【答案】(1) PUSH CX
 (2) RCL
 (3) BH
 (4) JNE SET_ONE
 (5) BH
 (6) AND BH, 01H
 (7) POP CX

【分析】

此程序是实现将 10 个无符号 8 位格雷码转换成 10 个 8 位二进制数的功能。关键是要处理好移位的过程。移位是为了处理每一位格雷码向二进制的转换。寄存器 AX 存放的是原格雷码，BX 存放处理后的二进制码。其中 AL 与 BL 分别存放处理前的格雷码以及处理后的二进制码；而 AH 和 BH 是用作异或处理的。

程序的堆栈段定义了一个 128 字节的堆栈区。堆栈段下面是数据段。数据段中定义了四个字节型变量，原始数据区 SOURCE、结果数据区 RESULT、原始数据文件名 NAME0、结果数据文件名 NAME1。数据段下面是代码段，告诉汇编程序，代码段、数据段和堆栈段分别属于段寄存器 CS、DS、SS。

AGN0 段是给一些寄存器如 CX、BX、AH 赋初值。第 (1)、(7) 空分别填写压入和弹出 CX。压入堆栈寄存器，是为了保存返回的地址，而弹出操作，是当程序结束后能够正确的返回操作系统。CX 是用于控制循环次数，压入堆栈后，CX 赋值为 8，因为每个数有 8 位格雷码，需要循环 8 次处理每一位码子；每次处理完一个数后，再从栈顶弹出 CX。

第 (2) 空及下一条指令都是把进位标志位 CF 包含在循环移位中的左循环移位指令。

这样就实现了将的 AL 最高位移入 CF, 然后 CF 又移入到 AH 的最低位。

第(3)空则是处理格雷码, 其中 AH 中存放的是当前的格雷码(如 g6), 而 BH 中存放的前一位二进制码(如 d7), 将它们进行异或操作, 就得到了 d6。

第(4)空考查的是异或操作, 若 AH 与 BH 中内容不同, 则异或结果为 1, 否则为 0。

第(5)空是将 BL 中二进制码子移入到 BH 中, 以便处理下一个格雷码。

第(6)空是将 BH 内容的高 7 位都屏蔽掉, 所以填写指令“AND BH, 01H”。因为每一位格雷码转换成二进制码时, 仅与前一个二进制码有关, 与其他的二进制码无关, 所以只留下当前所需要的 1 位二进制码。

接着, 将转换后的二进制码存入到 DI 当前所指向的地址单元(即[DI])中, SI 和 DI 地址各加 1, 循环处理下一个格雷码, 循环入口处是 AGN0。

★★

第 3 题

请编制程序, 其功能是: 内存中连续存放着 20 个十六位二进制无符号数序列, 请将它们排成升序(从小到大)。

例如:

内存中有 7001H, 7004H, 7002H... (假设后 17 个字均大于 7004H)

结果为 7001H, 7002H, 7004H... (后跟 17 个字, 按从小到大的顺序排列)

部分程序已给出, 其中原始数据由过程 LOAD 从文件 INPUT1.DAT 中读入 SOURCE 开始的内存单元中。运算结果要求从 RESULT 开始存放, 由过程 SAVE 保存到文件 OUTPUT1.DAT 中。

请填空 BEGIN 和 END 之间已给出的源程序使其完整(空白已用横线标出, 每个空白一般只需一条指令, 但采用功能相当的多条指令亦可), 或删除 BEGIN 和 END 之间原有的代码并自行编程来完成所要求的功能。

对程序必须进行汇编, 并与 IO.OBJ 链接产生可执行文件, 最终运行程序产生结果。调试中若发现整个程序中存在错误之处, 请加以修改。

试题程序:

```
                EXTRN     LOAD:FAR,SAVE:FAR
N                EQU      20

STAC            SEGMENT  STACK
                DB        128 DUP (?)
STAC            ENDS

DATA            SEGMENT
SOURCE          DW        N DUP(?)
RESULT          DW        N DUP(0)
NAME0           DB        'INPUT1.DAT',0
```

```

NAME1    DB          'OUTPUT1.DAT',0
DATA      ENDS

CODE      SEGMENT
          ASSUME      CS:CODE, DS:DATA, SS:STAC
START     PROC        FAR
          PUSH        DS
          XOR         AX,AX
          PUSH        AX
          MOV         AX,DATA
          MOV         DS,AX

          LEA         SI,SOURCE      ;数据区起始地址
          LEA         SI,NAME0      ;原始数据文件名
          MOV         CX,N*2        ;字数
          CALL        LOAD          ;从'DAT'中读取数据
;      **** BEGIN ****
          LEA         SI,SOURCE
          LEA         DI,RESULT
          MOV         CX,N
NEXT0:    MOV         AX,[SI]
          MOV         [DI],AX
          ADD         SI,____(1)____
          _____(2)_____
          LOOP        _____(3)_____
          CLD
          MOV         BX,N-1
MAL1:     LEA         SI,RESULT
          MOV         CX,____(4)____
NEXT:     LOD        _____(5)_____
          CMP         [SI],AX
          JAE         CONT
          XCHG        [SI],____(6)____
          MOV         [SI-2],AX
CONT:     LOOP        _____(7)_____
          _____(8)_____
          _____(9)_____      MAL1
;      **** END ****
          LEA         DX,RESULT      ;结果数据区首址
          LEA         SI,NAME1      ;结果文件名

```

```

MOV      CX,N*2      ;结果字数
CALL     SAVE        ;保存结果到文件
RET
START    ENDP
CODE     ENDS
END      START

```

【答案】(1) 2

(2) ADD DI, 2

(3) NEXT0

(4) BX

(5) SW

(6) AX

(7) NEXT

(8) DEC BX

(9) JNZ

【分析】

此程序是用冒泡排序法对一个含 20 个无符号数的序列实现升序排序的功能。冒泡排序方法的原理为：从第一个数据开始与相邻的数进行比较，若次序不对，两数交换位置。依次类推，直至第 $n-1$ 个数和第 n 个数进行过比较为止。上述过程称为第一趟冒泡排序。第一趟冒泡排序经过 $n-1$ 次比较后，最大的数已经到了数组尾，第二趟仅需比较 $n-2$ 次比较就够了，一共比较 $n-1$ 趟就完成了排序。这样共有两重循环：CX 是内循环次数，而 BX 是外循环次数。

程序的堆栈段定义了一个 128 字节的堆栈区。堆栈段下面是数据段。数据段中定义了两个字型变量：原始数据区 SOURCE、结果数据区 RESULT；两个字节型变量：原始数据文件名 NAME0、结果数据文件名 NAME1。数据段下面是代码段，告诉汇编程序，代码段、数据段和堆栈段分别属于段寄存器 CS、DS、SS。

第 (1)、(2) 空 SI 和 DI 地址要分别加 2，因为原始数据和结果数据都是以字型存放的。

接着开始循环，循环入口处是 NEXT0，所以第 (3) 空填写“NEXT0”。第 (3) 空之前的几条指令是实现将原始数据区 SOURCE 中的数据依次装入到 RESULT 中。

CLD 指令是将方向标志 DF 清 0，在数据操作时，实现地址指针自动增量。

MALI 段和 NEXT 段是实现每趟内的数据比较。第 (4) 空中第一趟比较 $N-1$ 次后，最大的数已到了末尾，第二趟仅需比较 $N-2$ 次就够了，所以 CX 是随着 BX 变化的。

第 (5) 空是需要将由 SI 作为地址指针的串元素装入到 AX 中，同时修改 SI，指向下一个串元素。

然后将下一个数与当前的数进行比较，若大于，则继续循环，取再一个数；若小于，则两数交换。所以第 (6) 空应该填写“AX”。

第 (7)、(8)、(9) 空则是完成一趟比较后，BX 应减 1，若 BX 不为 0，则继续循环，

进行第二趟比较,循环入口处是 NEXT。所以第(7)空填写“NEXT”,第(8)空填写“DEC BX”,第(9)空填写“JNZ”。

★★

第 4 题

请编制程序,其功能是:内存中连续存放着 24 个无符号二进制字序列,字的最高 3 位为 000,此序列对应某一信号在一段时间内的连续变化,现对第 21 个二进制字前的 20 个二进制字进行移动平均处理,其方法为:将要处理的字 X_i 用以它为开始的连续五个字的平均数 $(X_i+X_{i+1}+X_{i+2}+X_{i+3}+X_{i+4})/5$ 代替(余数舍去),得到新的 20 个无符号二进制字序列,结果存入内存。

例如:

内存中有 0100H, 0200H, 0300H, 0400H, 0500H, 0600H...

结果 0300H, 0400H...

部分程序已经给出,其中原始数据由过程 LOAD 从文件 INPUT1.DAT 中读入 SOURCE 开始的内存单元中,转换结果要求从 RESULT 开始存放,由过程 SAVE 保存到文件 OUTPUT1.DAT 中。

请填空 BEGIN 和 END 之间已经给出的一段源程序使其完整,需填空处已经用横线标出,每个空白一般只需填一条指令或指令的一部分(指令助记符或操作数),也可以填入功能相当的多条指令,或删去 BEGIN 和 END 之间原有的代码并自行编程来完成所要求的功能。

对程序必须进行汇编,并与 IO.OBJ 链接产生可执行文件,最终运行程序产生结果。调试中若发现整个程序中存在错误之处,请加以修改。

试题程序:

```

                EXTRN     LOAD:FAR,SAVE:FAR
N               EQU       24

STAC            SEGMENT  STACK
                DB        128 DUP (?)
STAC            ENDS

DATA            SEGMENT
SOURCE          DB        N DUP(?)           ;顺序存放 24 个字
RESULT          DB        20 DUP(0)          ;存放结果
NAME0           DB        'INPUT1.DAT',0
NAME1           DB        'OUTPUT1.DAT',0
DATA            ENDS

CODE            SEGMENT
                ASSUME     CS:CODE, DS:DATA, SS:STAC

```

```

START    PROC    FAR
          PUSH    DS
          XOR     AX,AX
          PUSH    AX
          MOV     AX,DATA
          MOV     DS,AX

          LEA     DX,SOURCE    ;数据区起始地址
          LEA     SI,NAME0     ;原始数据文件名
          MOV     CX,N*2       ;字节数
          CALL    LOAD         ;从'INPUT1.DAT'中读取数据
;    **** BEGIN ****
          MOV     DI,0
          MOV     SI,0
          MOV     CX,20
          MOV     BX,____(1)____
AGN0:     MOV     AX,SOURCE[SI]
          PUSH    SI
          PUSH    CX
          MOV     ____ (2) ____ , ____ (3) ____
AGN1:     INC     SI
          INC     SI
          ADD     AX,SOURCE[SI]
          LOOP    AGN1
          ____ (4) ____
          DIV     BX
          MOV     RESULT[DI],AX
          INC     DI
          ____ (5) ____
          POP     CX
          POP     SI
          INC     SI
          ____ (6) ____
          LOOP    AGN0
;    **** END ****
          LEA     DX,RESULT    ;结果数据区首址
          LEA     SI,NAME1     ;结果文件名
          MOV     CX,40        ;结果字节数
          CALL    SAVE         ;保存结果到文件

```



```

                RET
START          ENDP
CODE           ENDS
                END          START

```

【答案】(1) 5

(2) CX

(3) 4

(4) CWD

(5) INC DI

(6) INC SI

【分析】

此程序是处理 24 个无符号二进制字序列，其中前 20 个二进制字进行移动平均处理，将要处理的字用以它为开始的连续五个字的平均数代替，最后 4 个二进制字数据不变。要注意每个二进制字是占两个字节。所以，每次寻址时，数据区起始地址 SI 两次加 1，结果数据区地址 DI 也是两次加 1。

程序的堆栈段定义了一个 128 字节的堆栈区。堆栈段下面是数据段。数据段中定义了两个字型变量：原始数据区 SOURCE、结果数据区 RESULT；两个字节型变量：原始数据文件名 NAME0、结果数据文件名 NAME1。数据段下面是代码段，告诉汇编程序，代码段、数据段和堆栈段分别属于段寄存器 CS、DS、SS。

第(1)空中，由下面的程序语句可知，BX 是作为 AX 的除数，因为是求 5 个数的平均数，所以应该填“5”。

AHN0 段是将原始数据装入到 AX 中，其中 SI 是作为原始数据的偏移地址，其初值为 0。

第(2)、(3)空中 CX 是用来控制循环次数，因为是第 1 个数与后面的 4 个数相加，所以要循环四次，故 CX 赋值为 4。

第(4)空的前几条指令是将 SI 地址加 2，取下一个数据，与第一个数据相加。然后再从 AGN1 处开始循环，再将 AX 与后三个数相加。接着求平均数，将五个数的和除以 5，除法结果，商在 AX 中，余数在 DX 中。因为余数舍去，所以第(4)空是要给 DX 赋初值为 0。将求得平均数 AX 存入到[DI]中。

第(5)空是将 DI 地址再加 1，第(6)空是将 SI 地址再加 1，因为不管是原始数据还是结果数据，它们都是以字的形式存放的，两个字节一共占内存中的连续两个地址。

求完一个平均数后，接着开始求下一组五个数的平均数，循环从 AGN0 开始。

★★

第 5 题

请编制程序，其功能是：内存中从 SOURCE 开始连续存放着 21 个八位有符号数（补码），其相邻两数之间差值不超过 -8 至 7。对这种变化缓慢的数据可采用差分方法进行压缩。