

全国计算机等级考试

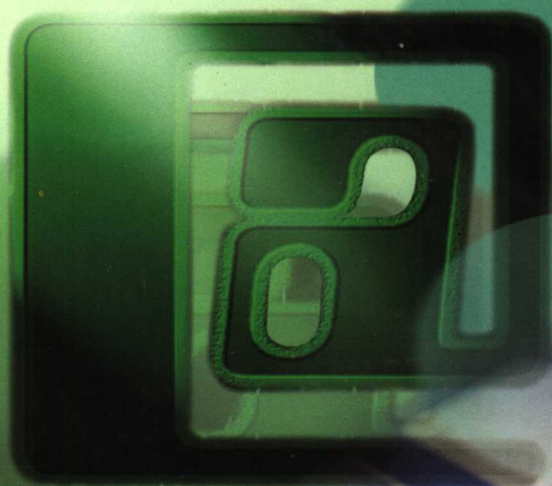


历届上机真题详解

三级汇编语言程序设计

(2008版)

全国计算机等级考试命题研究组 编
南开大学出版社



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: 2008 版. 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. —4 版. —天津: 南开大学出版社, 2007. 11

ISBN 978-7-310-02286-1

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(2007)第 154576 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

*

河北昌黎太阳红彩色印刷有限责任公司印刷

全国各地新华书店经销

*

2007 年 11 月第 4 版 2007 年 11 月第 4 次印刷

787×1092 毫米 16 开本 18.625 印张 466 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

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，并对其进行了详细准确的分析，指出考核的知识点、重点难点、解题思路。本书配套光盘主要包含两部分内容：（1）上机考试的全真模拟环境，本书所有试题均可在该环境中进行答题和评分，进行考前强化训练；（2）考试过程的录像动画演示，从登录、答题到交卷，均有指导教师的全程语音讲解。

本书针对参加全国计算机等级考试三级汇编语言程序设计（即三级 PC 技术）的考生，同时也可作为大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

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,我们做了大量市场调研工作,根据考生的备考体会,以及培训教师的授课经验,推出了《上机考试习题集——三级汇编语言程序设计》。本书主要由如下三部分组成。

一、上机考试题库及答案分析

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

本书提供了三级汇编语言程序设计的上机考试题库以及每道题的答案和详细分析。

二、上机考试必备知识

讲解上机考试的知识准备、如何复习和准备考试、考试注意事项以及上机考试过程。

三、配套光盘

本书配套光盘内容丰富,物超所值,可用于考前实战训练,主要内容有:

- 上机考试的全真模拟环境,用于考前实战训练。本上机系统题量巨大,书中所有试题,均可在全真模拟考试系统中进行训练和判分,以此强化考生的应试能力,其考题类型、出题方式、考场环境和评分方法与实际考试相同,但多了详尽的答案和解析,使考生可掌握解题技巧和思路。
- 上机考试过程的视频录像,从登录、答题到交卷的录像演示,均有指导教师的全程语音讲解。

为了保证本书及时面市和内容准确,很多朋友做出了贡献,陈河南、贺民、许伟、侯佳宜、贺军、于樊鹏、戴文雅、戴军、李志云、陈安南、李晓春、王春桥、王雷、韦笑、龚亚萍、冯哲、邓卫、唐玮、魏宇、李强等老师付出了很多辛苦,在此一并表示感谢!

在学习的过程中,您如有问题或建议,请使用电子邮件与我们联系。或登录百分网,在“书友论坛”与我们共同探讨。

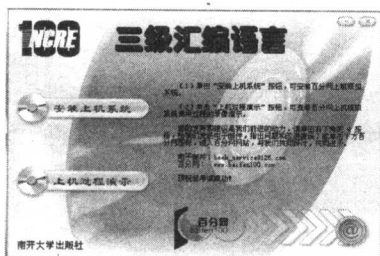
电子邮件: book_service@126.com

百分网: www.baifen100.com

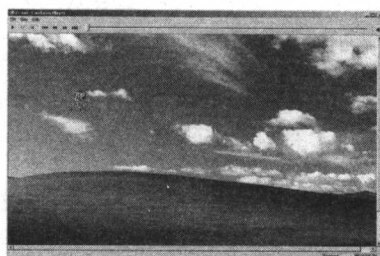
全国计算机等级考试命题研究组

2007年10月

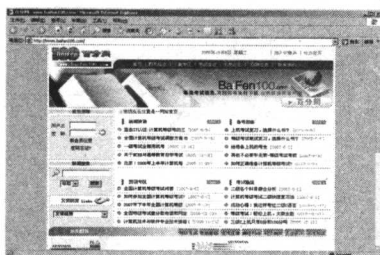
配套光盘说明



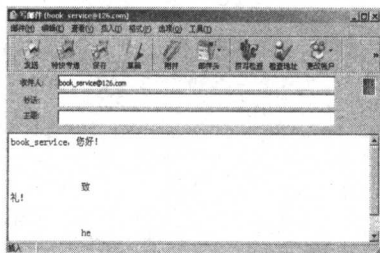
光盘初始启动界面, 可选择安装上机系统、查看上机操作过程



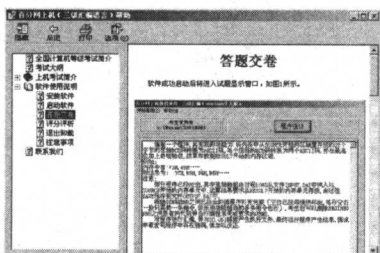
上机操作过程的录像演示, 有指导教师的全程语音讲解



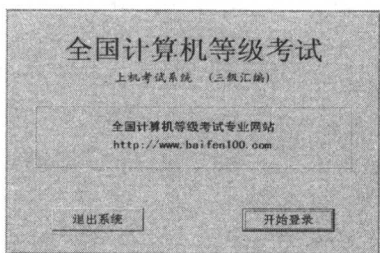
单击光盘初始界面的  图标, 可进入百分网, 您可以在此与我们共同探讨问题



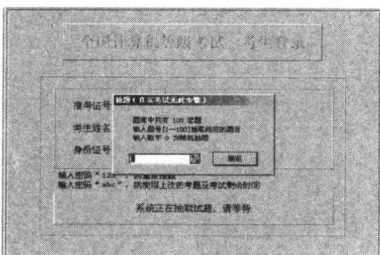
单击光盘初始界面左下角的  图标, 您可以给我们发送邮件, 提出您的建议和意见



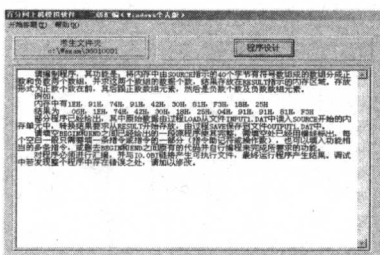
从“开始”菜单可启动帮助系统, 在这里可看到考试简介、考试大纲以及详细的软件使用说明



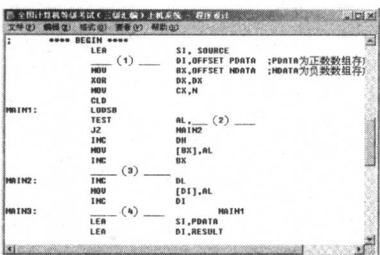
双击桌面上的软件名称启动上机系统, 进入登录窗口



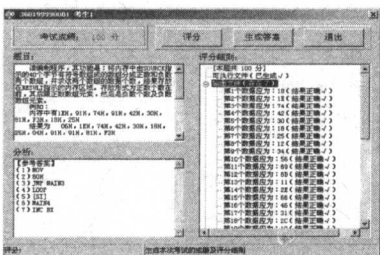
您可以随机抽题, 也可以指定固定的题目



浏览题目界面, 查看考试题目, 单击“考试项目”开始答题



实际答题环境。答题完成后单击工具栏中的“交卷”按钮



答案和分析界面, 查看所考核题目的答案和分析

目 录

第一部分 真题详解	1
第二部分 上机考试必备知识	278
2.1 应试知识准备	278
2.1.1 基本汇编指令	278
2.1.2 汇编程序的编制和调试	280
2.1.3 常用算法的汇编语言实现	281
2.2 复习和准备考试	281
2.2.1 复习阶段	281
2.2.2 熟悉考场环境	282
2.3 上机考试注意事项	282
2.3.1 上机考试要求	282
2.3.2 上机考试时间	283
2.3.3 等待评分结果	283
2.3.4 特殊情况处理	283
2.4 上机考试过程	283
2.4.1 登录过程	284
2.4.2 考试过程	285
2.4.3 交卷	286

第一部分 真题详解

★★

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 DUP (0)
PDATA     DB          N DUP (0)
DATA       ENDS
```

```

CODE    SEGMENT
        ASSUME    CS:CODE, DS:DATA, SS:STAC
START    PROC
        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, SOURCE
        (1)       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, (2)
        JZ        MAIN2
        INC       DH
        MOV       [BX], AL
        INC       BX
        (3)
MAIN2:   INC      DL
        MOV       [DI], AL
        INC       DI
MAIN3:   (4)      MAIN1
        LEA       SI, PDATA
        LEA       DI, RESULT
        MOV       [DI], DL
        INC       DI
        XOR       CX, CX
        MOV       CL, DL
MAIN4:   MOV      AL, (5)
        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

【解析】

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。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 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

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到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 $d_k(k=6, \dots, 0)$ 分别为格雷码的位 $g_k(k=6, \dots, 0)$ 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

        STAC      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       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       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)
        RCL       AH,1
        CMP       AH,(3)
        (4)
        JMP       NEXT
SET_ONE: OR       BL,01H
NEXT:    MOV       (5),BL
        (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

```

(7) POP CX

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         DX, 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'INPUT1.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

【解析】

此程序是用冒泡排序法实现一个含 20 个无符号数的序列的升序排序的功能。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。

MAL1 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  DW       N DUP(?) ;顺序存放 24 个字
RESULT  DW       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

【解析】

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”。

AH0 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。