计算机及应用专业自学考试同步辅导丛书

汇编语言程序设计应试指导

（专科）

苏 光 奎　　编著

清 华 大 学 出 版 社

北　京

内 容 简 介

本书是全国高等教育自学考试指定教材《汇编语言程序设计》(计算机及应用专业——专科)的同步辅导用书，完全遵循自学考试大纲的要求，总结出汇编语言程序设计课程的知识体系和要点，针对重点与难点设计典型例题并细致地分析，并通过大量模拟试题的练习进行强化。

本书共分 7 章，第 1 章介绍了计算机系统的基本组成、8086/8088 CPU 的寄存器组中各寄存器的作用和 8086/8088 CPU 的存储器组织；第 2 章讲述了 8086/8088 的寻址方式和指令系统；第 3 章详细介绍了汇编语言程序设计中语句的格式和伪指令；第 4 章至第 7 章分析了顺序程序设计、分支程序设计、循环程序设计的基本方法和设计技巧。每章均包括“学习目的和要求”、“知识体系”、“例题分析”、“练习题及参考答案”4 个部分，条理清晰、重点明确、为广大读者学习和巩固汇编语言程序设计课程知识提供了一条很好的学习捷径。

本书的特色是在尽可能覆盖全部考核内容的基础上，加强对考核重点与难点的分析与练习，可供参加高等教育计算机及应用专业自学考试的考生作为复习参考，也可作为自学考试辅导教师的教学参考用书。

图书在版编目（CIP）数据

汇编语言程序设计应试指导（专科）/ 苏光奎编著.
北京：清华大学出版社，2003
（计算机及应用专业自学考试同步辅导丛书）
ISBN 7-302-06671-X

Ⅰ. 汇… Ⅱ. 苏… Ⅲ. 汇编语言-程序设计-高等教育-自学考试-自学参考资料 Ⅳ.TP313

中国版本图书馆 CIP 数据核字（2003）第 039554 号

出版者：清华大学出版社（北京清华大学学研大厦，邮编 100084）
http://www.tup.com.cn
印刷者：北京市耀华印刷有限公司
发行者：新华书店总店北京发行所
开 本：787×1092 1/16 印张：18.375 字数：446 千字
版 次：2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷
书 号：ISBN 7-302-06671-X/TP・4993
印 数：0001~5000
定 价：23.00 元

目　　录

丛　书　序

为了适应社会主义现代化建设的需要，我国于 1981 年开始实行高等教育自学考试制度。它是个人自学、社会助学和国家考试相结合的一种教育形式，是高等教育的有机组成部分，其职责就是在高等教育这个水平上倡导自学、鼓励自学、帮助自学、推动自学，为每一位自学者铺就成才之路。20 余年来，高等教育自学考试以其严格的质量和良好的声誉得到了社会的普遍关注，近千万的考生通过自学考试获得了本科、大专和中专学历文凭。

随着计算机技术在我国各个领域的推广和普及，越来越多的行业与单位把操作和应用计算机作为劳动者必须掌握的一种基本技能。许多单位已把掌握一定的计算机知识和应用技能作为干部录用、职务晋升、职称评定、上岗资格的重要依据。故近年来参加计算机及应用专业自学考试的考生越来越多。

计算机行业是一个发展迅猛的行业，技术在不断进步，社会需求也在不断地随之变化，因而自学考试大纲也进行了若干调整，国家教育部考试中心从 2000 年开始，正式执行自学考试新计划，同时施行新编的大纲和教材。虽然新编自学考试教材适合自学，有利于学习者培养实践意识，提升自学能力，但仍无法满足广大应试人员成功通过考试的迫切需要。

为了满足广大自学应考者的学习、复习和应试的要求，北京科海培训中心精心策划了这套“计算机及应用专业自学考试同步辅导丛书”。本套丛书包括：

- 计算机网络与通信应试指导（本科）
- 计算机应用技术应试指导（专科）
- 数据库及其应用应试指导（专科）
- 数据库原理应试指导（本科）
- 计算机网络技术应试指导（专科）
- 数据结构应试指导（本科）
- 数据结构导论应试指导（专科）
- 汇编语言程序设计应试指导（专科）
- 面向对象程序设计应试指导（本科）
- 计算机组成原理应试指导（专科）
- 计算机系统结构应试指导（本科）
- 操作系统概论应试指导（专科）
- 操作系统应试指导（本科）

丛 书 特 点

本套丛书紧扣国家教育部考试中心最新颁布的考试大纲，以指定教材为基础，由长期工作在教学一线的教授、副教授、讲师亲自编写，从结构设计、内容安排到实例、练习题都经过精心设计与整理。丛书具有以下特点：

- 以考试大纲的各项要求和各章的考核知识点为主线，梳理学习要点，归纳知识体系。
- 注重基础、突出重点，以便考生对课程内容建立一个整体的概念。
- 深入浅出，条理清晰，语言通俗易懂。
- 注意对学生解题能力的培养，书中详细分析了大量的例题，并通过大量的针对性练习来强化对考核重点与难点的理解与应用。

编写过程中，严格按照指定教材的章节顺序安排内容。每一章首先列出总体要求、学习重点和难点，让读者做到心中有数，明白学习这一章要达到什么样的目标，什么是难点，什么是重点，特别要注意哪些地方。然后分知识体系、例题分析、练习题及参考答案 3 部分介绍。知识体系开宗明义，先给出知识体系结构图，让读者从整体上全面把握篇章结构，了解各部分之间的联系，复习起来思路明确、条理清晰；接下来是知识点与考核要求对重点内容进行适当讲解，并提出对此知识点应达到的能力层次。例题分析通过典型例题的分析和解答使学生在掌握基本概念的同时，进一步加深对内容的综合理解和应用。练习题与参考答案覆盖全部考核内容，同时加大重点内容的覆盖密度，习题类型与考试要求有关，包括填空题、选择题、简答题和算法设计题。

使 用 说 明

本丛书是与高等教育自学考试指定教材配套使用的同步辅导用书，知识体系部分突出强调了考试重点，例题和练习题部分则覆盖了全部考核内容，还包含了指定教材中的部分课后习题。例题和练习题部分涉及的个别概念在本书知识体系部分可能未曾提及，所以最好与指定的教材配套使用本书。

前　言

随着计算机技术的发展，对原有计算机专业的教学模式提出了挑战，同时也带来了前所未有的机遇。为了为国家培养高素质的跨世纪科技人才，也为了满足读者的需要，希望在学习相关课程中能进一步加深对相关课程知识的理解，计算机专业相关课程的辅导教材是已成为广大学者不可缺少的学习资料。

“汇编语言程序设计”是我国高等学校计算机及应用专业的一门主干课程，也是电子信息、自动控制、信息管理等专业的重要基础课程。该课程是从事计算机研究与应用，特别是从事软件研究的基础课程。该课程从系统软件和应用软件的角度出发，以目前最为广泛的 IBM-PC 机为例，详细介绍了宏汇编语言的基本概念、基本原理和程序设计的基本方法。

“汇编语言程序设计应试指导”在介绍“汇编语言程序设计”课程基本知识的基础上，以例题分析的形式对知识点进行了归纳、总结。同时，以单项选择题、多项选择题、填空题、判断题、改错题、判断改错题、简答题、程序分析题、程序填空题、程序设计题的形式向读者提供练习题，并附有参考答案，使读者能对“汇编语言程序设计”的基本概念、基本知识做进一步的理解，同时也训练了学生的编程能力。所以，“汇编语言程序设计应试指导”为读者学习“汇编语言程序设计”课程提供了一个巩固、深化课堂知识的环节，提高了学生分析问题、解决问题的能力。

“汇编语言程序设计应试指导”共分 7 章，第 1 章介绍了计算机系统的基本组成、8086/8088 CPU 的寄存器组中各寄存器的作用和 8086/8088 CPU 的存储器组织；第 2 章讲述了 8086/8088 的寻址方式和指令系统；第 3 章详细介绍了汇编语言程序设计中语句的格式和伪指令；第 4 章至第 7 章分析了顺序程序设计、分支程序设计、循环程序设计的基本方法和设计技巧。

本书适宜于高等院校、高职高专、成人教育类计算机专业或相关专业的教学辅助用书，也可供科研和软件开发人员的学习参考用书。

由于作者的水平有限，再加之时间仓促，书中难免有错误之处，诚请读者批评指正。

作　者

2003.05

第 1 章　基 础 知 识

1.1 学习目的和要求

汇编语言是一种面向机器的语言。学习汇编语言程序设计是软件设计的需要，是阅读、掌握和改进现有的系统软件和应用软件的需要，是对计算机进行软、硬件维护的需要。汇编语言程序设计是其他高级语言程序设计的基础，它也对计算机组成原理、微机原理、微机接口技术等的学习有着决定性的影响。

在本章的学习中，要求能正确地理解学习汇编语言程序设计的目的和重要性；正确地理解并熟练地掌握有关计算机系统的基本组成；了解汇编语言程序设计的特点和作用；掌握机器语言、汇编语言、汇编源程序、汇编程序、汇编等基本概念；掌握 8086/8088 CPU 的组成、8086/8088 CPU 的寄存器组中各寄存器的作用和 8086/8088 CPU 的存储器组织。

【学习重点】

1. 计算机系统的基本组成。
2. 8086/8088 CPU 的寄存器组中各寄存器的作用。
3. 8086/8088 CPU 的存储器组织。

【学习难点】

8086/8088 CPU 的存储器组织。

1.2 知识体系

1.2.1 知识体系结构

在本章中，所讲述的知识体系结构如图 1.1 所示。

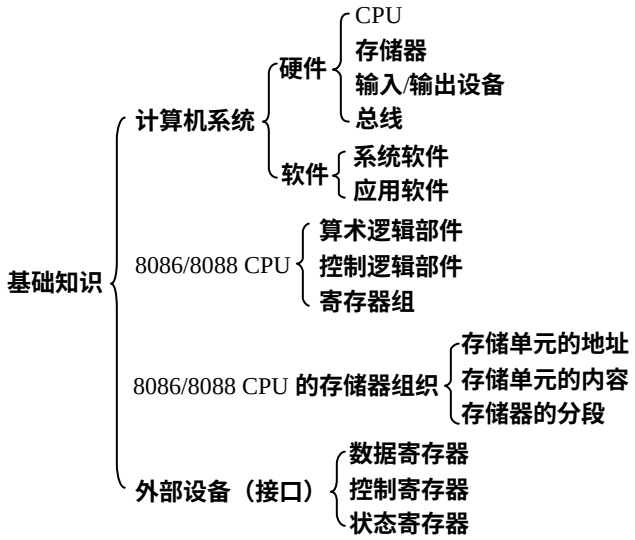


图 1.1　基础知识体系结构

1.2.2　知识点与考核要求

1. 计算机系统的基本组成，应达到“识记”层次。

（1）计算机系统的基本组成。
（2）硬件系统的组成及各部分的功能。
（3）软件系统的组成及汇编语言、汇编源程序、汇编程序、目标程序、连接程序、调试程序的基本概念。

2. 8086/8088 汇编语言编程的硬件模型，应达到“综合应用”的层次。

（1）8086/8088 CPU 内部数据寄存器组和寄存器组中各寄存器的名称、符号、位数和功能。
（2）8086/8088 CPU 中的 IP 寄存器的位数和功能。
（3）8086/8088 CPU 中的 PSW 的状态标志位和控制标志位的名称、符号、功能和状态符号的表示。
（4）8086/8088 CPU 的组成形式和特点、存储器地址的分段、存储单元字节和字的读写。

3. 汇编语言程序设计的特点和作用，应达到“识记”的层次。

（1）学习汇编语言程序设计的目的和意义。
（2）汇编语言程序设计的特点和作用。

1.3 例题分析

例 1　在取指令时，使用的段寄存器一般是（　　）。

解：因为 8086/8088 CPU 有 4 个段寄存器，DS 是指向数据段，SS 是指向堆栈段，ES 是指向附加段，CS 是指向代码段。而程序经汇编后产生的机器代码是存放代码段的，所以程序一般存放在代码段（即 CS）。

例 2　若执行加法操作前，AL=87H，BL=92H，两个数据执行加法操作后，CF 应为（　　），OF 应为（　　）。

解：AL 中的内容和 BL 中的内容相加，执行

$$\begin{array}{rr} & 1000\ 0111 \\ + & 1001\ 0010 \\ \hline 1\leftarrow & 0001\ 1001 \end{array}$$

的操作。由于 87H 和 92H 相加时，其最高位产生进位，所以 CF=1；而 8 位二进制数的表示范围是+127~-128，两个负数相加后得到的结果为正，所以 OF=1。

对于此类判断溢出的方法有两种。一种方法是同号相加（或者异号相减）时，其结果的符号与被加数（或被减数）的符号相反，则 OF=1；否则 OF=0。如本例中的两个负数相加得到的结果为正，其被加数的符号与结果符号相反，所以 OF=1。另一种方法是最高位的进位位与次高位的进位位异或的结果就是 OF 的值。如本例中的最高位进位位为 1，次高位进位位为 0，两个进位位相异或 1⊕0=1，所以 OF=1。

例 3　若 CS=1000H，IP=0200H，则下一条指令的物理地址为（　　）。

解：8086/8088 CPU 形成内存物理地址的过程是将段地址左移 4 位加偏移地址。若要形成下一条指令的物理地址，则应选用当前代码段。将代码段 CS 寄存器的值 1000H 左移 4 位得 10000H，其偏移地址为 IP 寄存器的值 0200H，所以代码段 CS 寄存器的值左移 4 位加偏移地址 IP 寄存器的值得到下条指令的物理地址为 10200H。

例 4　在 8086/8088 CPU 中，一个段最大可定义的字节数为（　　）。

A. 16K　　B. 32K　　C. 64K　　D. 1M

解：8086/8088 CPU 有 20 位的地址，它的最大内存空间为 1M 字节。而 CPU 内部寄存器一般是 16 位的（除 8 个 8 位的数据寄存器外），16 位寄存器最大的表示范围是 0~65535。如果直接用 16 位的寄存器来指示内存的某一单元是不能寻址 1M 字节的内存空间，所以采用了将段地址左移 4 位再加上偏移地址形成物理地址的方法。由于在寻址某一段内的某一地址时，其段地址是不变的，只改变偏移地址。而偏移地址是 16 位的，2 的 16 次方为 64K，所以应选 C。

例 5　如果内存中某一存储单元的物理地址是 12340H，偏移地址是 0200H，那么它的段地址是（　　）。

A. 12140H　　B. 10340H　　C. 1034H　　D. 1214H

解：逻辑地址是由段地址和偏移地址来表示的，而 20 位的物理地址是惟一的表示某一存储单元的地址。如果将物理地址 12340H 减去偏移地址 200H，求得 12140H。再将此值右移 4 位得到段地址 1214H。所以应选 D。

例 6　下列四个数均为有符号数，其中最大的数是（　　）。

A. 3274H　　B. 8365H　　C. 9564H　　D. 5342H

解：若一个数是有符号数，则最高位为 1 表示负数，最高位为 0 表示是正数。上面四个数中，由于选项 B 和选项 C 的最高位都是 1，所以是负数。选项 A 和选项 D 最高位是 0，都是正数。由此可知选项 A 和选项 D 都大于选项 B 和选项 C，而选项 D 又大于选项 A。所以应选 D。

例 7　有两个 16 位的字单元 1234H 和 5678H，依次存放在 IBM PC 机的存储器 10000H 开始的四个单元中，请用示意图表示它们在存储器的存放形式。

解：在 IBM PC 机中，存储器是按字节编址的。一个字节占用一个存储单元。如果是字节，就存放在对应单元；如果是字，则字的高 8 位存放在高地址单元，字的低 8 位存放在低地址单元。由此，1234H 的低 8 位 34H 应存放在 10000H 单元，高 8 位 12H 应存放在 10001H 单元。5678H 的低 8 位 78H 应存放在 10002H 单元，高 8 位 56H 应存放在 10003H 单元。存放示意图如图 1.2 所示。

10000H	34H
10001H	12H
10002H	78H
10003H	56H

图 1.2　数据存放示意图

例 8　8086/8088 CPU 按 8 位 I/O 端口地址进行寻址，其寻址范围是（　　）。

A. 16K　　B. 32K　　C. 64K　　D. 1M

解：由于 I/O 端口地址的寻址方式有两种。一是直接寻址方式，它在指令中直接给出端口的地址，其端口地址范围只能是 0～255。二是间接寻址方式，它由 DX 寄存器中的内容作为端口地址，其地址范围是 0～65535。所以端口地址的寻址范围最大是 64K，应选 C。

1.4 练习题与参考答案

1.4.1 单项选择题

1. 在下列的选项中，能够组成计算机系统的是（　　）。

A. 硬件系统和软件系统　　B. CPU、存储器、输入输出设备
C. 操作系统、各种语言　　D. 系统软件和应用软件

2. 汇编语言属于（　　）。

A. 用户软件　　B. 系统软件　　C. 高级语言　　D. 机器语言

3. 汇编语言源程序经汇编程序汇编后产生的文件的扩展名是（　　）。

A. EXE　　B. OBJ　　C. ASM　　D. LST

4. 汇编语言的（　　）文件经连接（LINK）后产生可执行文件。

A. ASM　　B. EXE　　C. LST　　D. OBJ

5. 中央处理器 CPU 是由（　　）组成的。

A. 运算器　　B.控制器　　C. 寄存器组　　D.前三者

6. IBM PC 机的 DOS 属于（　　）。

A. 用户软件　　B.系统软件　　C. 系统硬件　　D. 一种语言

7. 汇编语言源程序是（　　）程序。

A. 不可直接执行的　　B. 可直接执行的
C. 经汇编程序汇编后就可执行的　　D. 经连接后就可直接执行的

8. 8086/8088 CPU 的寄存器组中，8 位的寄存器共有（　　）个。

A. 4　　B. 6　　C. 8　　D. 10

9. 8086/8088 CPU 的寄存器组中，16 位的寄存器共有（　　）个。

A. 10　　B. 12　　C. 13　　D. 14

10. 8086/8088 CPU 执行算术运算时 PSW 共有（　　）个标志位会受影响。

A. 4　　B. 5　　C. 6　　D. 7

11. 在程序执行过程中，IP 寄存器始终保存的是（　　）。

A. 上一条指令的首地址　　B. 本条指令的首地址
C. 下一条指令的首地址　　D. 需计算有效地址后才能确定的地址

12. IBM PC 机的存储器可分（　　）个段。

A. 4　　B. 256　　C. 512　　D. 65536

13. 当使用 BP 作编程地址时，此时使用的是（　　）段。

A. CS　　B. DS　　C. ES　　D. SS

14. 如果指令的运算结果为 0，则（　　）标志位为 1。

A. SF　　B. OF　　C. ZF　　D. CF

15. 如果指令的运算结果中有奇数个 1，则（　　）标志位为 0。

A. PF　　B. CF　　C. OF　　D. SF

16. IBM PC 机的内存是按（　　）编址的。

A. 位　　B. 字节　　C. 字　　D. 双字

17. 存储器的一个字节表示（　　）位。

A. 8　　B. 16　　C. 32　　D. 64

18. 如果某个字变量的数据存放在奇地址单元，则 8086/8088 CPU 读写该变量时需要（　　）个读写周期。

A. 1　　B. 2　　C. 3　　D. 4

19. 在机器内部操作中，CPU 与存储器之间的任何信息交换使用的都是（　　）地址。

A. 逻辑　　B. 物理　　C. 有效　　D. 相对

20. 一个 16 位相对位移的范围是（　　）。

A. 0~65535　　B. 0000H~FFFFH
C. 8000H~7FFFH　　D. 8000H~FFFFH

21. 物理地址的形成过程是将段地址左移（　　）位加偏移地址。

A. 2　　B. 3　　C. 4　　D. 5

22. 如果内存的某一单元的逻辑地址为 236FH：1000H，则物理地址为（　　）。

A. 1236FH　　B. 336FH　　C. 336F0H　　D. 246F0H

23. 如果某一存储单元的物理地址为 12345H，则它的逻辑地址为（　　）：0345H。

A. 0012H　　B. 12000H　　C. 1200H　　D. 0120H

24. 如果一个字变量中存放 16 个字，该字变量的起始地址为 1000H：2000H，则该字变量数据区中的最末一个字单元的物理地址为（　　）。

A. 1201FH　　B. 12000H　　C. 1201EH　　D. 12020H

25. 设 22000H、22001H 单元分别存放的数据为 12H、34H，若要读取 22000H 字单元中的数据，此时读出的数据是（　　）。

A. 12H　　B. 34H　　C. 3412H　　D. 1234H

26. 如果数据存放在以 DI 的内容为偏移地址的数据段中，设 DS＝3624H，DI＝2200H，则此存储单元的物理地址为（　　）。

A. 38440H　　B. 58240H　　C. 25624H　　D. 58240H

27. 一般的外部设备接口电路中的状态寄存器是用来存放外设或接口电路的（　　）信息。

A. 数据　　B. 控制　　C. 状态　　D. 前三者

28. 下列叙述正确的是（　　）。

A. 不同类型的机器字长是一样的　　B. 一个字节有 8 位二进制数
C. 各种不同的机器指令系统都是一样的　　D. 机器指令都是 8 位的

29. 下列叙述正确的是（　　）。

A. 8088 CPU 的内部数据总线是 8 位的，外部数据总线是 8 位的
B. 8088 CPU 的内部数据总线是 16 位的，外部数据总线是 16 位的
C. 8086 CPU 的内部数据总线是 8 位的，外部数据总线是 8 位的
D. 8086 CPU 的内部数据总线是 16 位的，外部数据总线是 16 位的

30. 下列叙述正确的是（　　）。

A. 机器指令是可执行指令　　B. 汇编语言源程序可直接执行
C. 汇编语言目标程序可直接执行　　D. 高级语言程序可直接执行

单项选择题参考答案

1. A	2. B	3. B	4. D	5. D	6. B
7. A	8. C	9. D	10. C	11. B	12. D
13. D	14. C	15. A	16. B	17. A	18. B
19. B	20. C	21. C	22. D	23. C	24. C
25. C	26. A	27. C	28. B	29. D	30. A

1.4.2　多项选择题

1. 中央处理器 CPU 是由（　　）组成的。

A. 运算器　　B. 控制器　　C. 寄存器组　　D. 存储器

2. 下列寄存器中属于段寄存器的有（　　）。

A. SI　　B. DI　　C. SS　　D. ES

3. 下列属于系统软件的有（　　）。

A. 操作系统　　B. 汇编语言　　C. 高级语言　　D. 编译程序

4. 下列的 16 位寄存器中能够用 2 个 8 位寄存器来表示的有（　　）。

A. SI　　B. AX　　C. BP　　D. BX

5. 逻辑地址是由（　　）组成的。

A. 段地址　　B. 物理地址　　C. 偏移地址　　D. 实际地址

6. 在下列的标志位中，不能根据运算结果置位/复位的标志位有（　　）。

A. IF　　B. PF　　C. TF　　D. OF

7. 在下列的标志位中，能够根据运算结果置位/复位的标志位有（　　）。

A. ZF　　B.PF　　C. DF　　D. SF

8. 在下列逻辑地址中，用来表示同一个物理地址 3240AH 的有（　　）。

A. 3040H：200AH　　B. 240AH：3000H
C. 3200H：040AH　　D. 3000H：240AH

9. 在下列文件中，能够在计算机上直接运行的有（　　）。

A. EXE 文件　　B. COM 文件
C. OBJ 文件　　D. BAT 文件

10. 在外部设备接口电路中，一般有（　　）。

A. 数据寄存器　　B.状态寄存器
C. 标志寄存器　　D. 控制寄存器

多项选择题参考答案

1. ABC	2. CD	3. ABCD	4. BD	5. AC
6. AC	7. ABC	8. ACD	9. AC	10. ABD

1.4.3 填空题

1. 中央处理器 CPU 包括_____，_____和内部可编程的_____。

答：运算器　控制器　寄存器组

2. 计算机的硬件系统包括_____，_____和输入输出子系统三个主要组成部分。
答：CPU　存储器

3. 汇编语言属于_____软件。
答：系统

4. 汇编语言源程序必须经过_____后再_____才能产生_____文件。
答：汇编程序汇编　连接　可执行（EXE）

5. 装入程序的作用是把要执行的_____和库文件或其他已翻译过的_____连接在一起形成机器能_____的程序。
答：程序　子程序　执行

6. 装入程序的作用是将程序从_____传送到_____。
答：外存储器　内存

7. 在 PC 机中，一些简单的汇编语言可以通过_____来建立、修改和执行。
答：调试程序

8. 操作系统属于_____软件。
答：系统

9. 调试程序属于_____软件。
答：系统

10. 用汇编语言编写的图像显示软件属于_____软件。
答：用户（应用）

11. 8086/8088 CPU 的地址有_____位。可寻址的内存地址空间是_____。
答：20　1MB

12. 8086/8088 CPU 有_____个段寄存器。
答：4

13. 在读取下一条指令时，_____中的内容作为该指令的段地址，_____中的内容作为该指令的偏移地址。
答：CS，IP

14. 当两个无符号数的字节相加时，其结果超过_____时就将 CF 置 1。
答：255

15. 当两个有符号数的字节相加时，其结果超过_____时就将 OF 置 1。
答：-128 ~ +127

16. 若运算的结果为负数，则 SF 为_____。
答：1

17. IBM PC 机的存储器是按_____编址的。

答：字节

18. 若某双字变量中存放两个数据，它占用_____个存储单元。

答：8

19. 在 8086/8088 CPU 的微机中，CPU 访问奇地址字单元需_____个内存读写周期，访问偶地址字单元需_____个内存读写周期。

答：2　1

20. IBM PC 机的存储器按段划分，每个段的大小可在_____范围内选取任意个_____数。

答：64KB　字节

21. IBM PC 机的 1MB 内存空间最多可分为_____个段。

答：64K

22. 逻辑地址是由_____和_____组成的，它可以形成 20 位的物理地址。

答：段地址　偏移地址

23. 形成内存物理地址的过程是将_____左移_____位加_____ 。

答：段地址　4　偏移地址

24. 某微机的字长为 32 位，一个字需占用内存_____个字节单元。

答：4

25. 若某存储单元的逻辑地址为 1200H：3400H，则该存储单元的物理地址为_____。

答：15400H

26. IBM PC 机可寻址的 I/O 端口地址有_____，共使用_____条地址线。

答：64K　16

27. 若要将字 1234H 存放在以 10000H 开始的存储单元之中，则 10000H=_____，10001H=_____。

答：34H　12H

28. 若要将双字 12345678H 存放在以 20000H 开始的存储单元之中，则 20002H=_____。

答：34H

29. 外设与接口电路中的控制寄存器中的内容是由_____送来的。

答：CPU

30. 外设与主机互递信息是通过外设接口电路实现的。一般的接口电路中有_____、_____和_____三种不同类型的寄存器。

答：数据寄存器　控制寄存器　状态寄存器

31. 压缩型 BCD 码一个字节能表示_____位十进制数，非压缩型 BCD 码一个字节能表示_____位十进制数。

答：2　1

32. 辅助进位位 AF 主要是用于对_____数据进行算术的调整。

答：十进制

1.4.4　简答题

1. IBM PC 微型计算机一般采用什么总线结构形式？它将哪几个功能部件通过总线连接在一起构成微型计算机的硬件系统？

【解答】IBM PC 微型计算机一般采用单总线的结构形式，它将 CPU、存储器和输入输出接口等连接在一起。

2. 8086/8088 CPU 为什么只能寻址 1MB 的内存空间？

【解答】因为 8086/8088 CPU 的地址线只有 20 位。2 的 20 次方可表示 1M。又因为 8086/8088 CPU 的微机是按字节编址的，所以它只能寻址 1MB 的内存空间。

3. IBM PC 机的存储器为什么要分段？怎样采用分段寻址？

【解答】因为 IBM PC 机可寻址 1MB 的内存空间，而 8086/8088 CPU 的内部寄存器只有 16 位的。16 位的寄存器只能表示 64K，它不能直接对 1MB 的内存空间进行寻址。为了寻址 1MB 的任何一个存储单元，必须将 1MB 的内存空间进行分段。分段的方法是将 1MB 的内存空间分为若干段，每个存储单元由段地址和段内的偏移地址表示，将段地址左移 4 位加偏移地址形成 20 位的物理地址。

4. 一个 8 位数能表示的最大值和最小值是多少？一个 16 位数能表示的最大值和最小值是多少？

【解答】当一个 8 位数据为无符号数时，它的表示范围是 0~255；当它为有符号数时，它的表示范围是-128~127。当一个 16 位数据为无符号数时，它的表示范围是 0~65535；当它为有符号数时，它的表示范围是-32768~32767。

5. 如何实现 ASCII 码数字字符与 BCD 码之间的相互转换？

【解答】将数字字符的 ASCII 码减去 30H 就可得到 BCD 码。

6. 简述存储器的逻辑地址、物理地址和有效地址。

【解答】用段地址和偏移地址表示存储单元的地址为逻辑地址。逻辑地址不是惟一的，同一地址可以有不同的表示。用 20 位的二进制表示存储单元的地址称为物理地址。每一个地址都是惟一的。由逻辑地址形成物理地址的过程是将段地址左移 4 位加偏移地址。计算偏移地址是由相关项组成的。由这些相关项计算得到的地址称为有效地址（EA）。

7. 堆栈操作的原则是什么？堆栈操作的过程是怎样进行的？

【解答】堆栈的操作是采用后进先出的原则。堆栈操作的过程是：在压栈时，先将 SP-2→SP，然后将要压入的字压入到 SS：SP 所指示的单元中；在弹栈时，先将 SS：SP 所指示的字单元中的内容弹出来送到目标地址中，然后将 SP+2→SP。

8. 在 8086/8088 系统中，内存的逻辑地址是由哪两部分组成的？

【解答】逻辑地址是由段地址和偏移地址两部分组成的。CS 寄存器中的内容作段地址时，IP 寄存器中的内容作偏移地址；SS 寄存器中的内容作段地址时，SP 寄存器中的内容作偏移地址。在没有段超越前缀的情况下，如果 BP 中的内容作操作数的基址指针，使用的段寄存器也是 SS。DS 作为寻找操作数的段寄存器，与它对应的寄存器有 BX、SI 和 DI。在串操作指令中，DI 中的内容作源操作数的偏移地址时，ES 作为段寄存器。在有段超越前缀时，则根据指定的段寄存器作段地址。

9. 下列操作可使用哪些寄存器？

（1）加法和减法。
（2）循环计数。
（3）乘法和除法。
（4）指示程序已执行到哪条指令的地址。
（5）指示当前从堆栈中弹出的数据的地址。
（6）表示运算结果为 0。

【解答】（1）AX、BX、CX、SI、DI、BP、SP
（2）CX
（3）AL、AX、DX
（4）CS：IP
（5）SS：IP
（6）PSW 的 ZF 位

10. 在存储器中存放的数据如图 1.3 所示，读出 23004H 字节单元中的内容是多少？读出 23002H 字单元中的内容是多少？读出 23003H 字单元中的内容是多少？23000H 双字单元之中的数据是多少？

	……
23000H	41H
23001H	37H
23002H	62H
23003H	5FH
23004H	47H
23005H	36H
	……

图 1.3　存储器存放的数据

【解答】47H，5F62H，475FH，5F623741H

11. 设 SS=1200H，堆栈压入 10 个字节后，SP=00F6H。请指出堆栈底部字单元的物理地址，堆栈顶部字单元的物理地址。

【解答】堆栈顶部字单元的物理地址为 12000H + 00F6H = 120F6H
堆栈底部字单元的物理地址为 12000H + 00F6H +（10 - 2）= 1200FEH

12. 设有一个 30 个字的数据区，它的起始地址是 2000H：3000H，请给出这个数据区的首、末字单元的物理地址。

【解答】首字单元的物理地址为 20000H + 3000H = 23000H
末字单元的物理地址为 20000H + 3000H +（30 - 1）* 2 = 2303AH

13. 请将下列左边的项与右边的解释联系起来（把所选字母填在括号中）

（1）CPU　（　）A. 保存当前的栈顶地址的寄存器。
（2）存储器　（　）B. 指示下一条要执行的指令的地址。
（3）堆栈　（　）C. 存储程序、数据等信息的记忆装置，PC 机有 ROM 和 RAM 两种。
（4）IP　（　）D. 以后进先出方式工作的存储空间。
（5）SP　（　）E. 把汇编语言程序翻译成机器语言程序的系统程序。
（6）状态标志　（　）F. 惟一代表存储空间中每个字节单元的地址。
（7）控制标志　（　）G. 能够被计算机直接识别的语言。
（8）段寄存器　（　）H. 用指令助记符、符号地址、标号等符号书写的程序语言。
（9）物理地址　（　）I. 把若干个程序模块连接起来成为可执行文件的系统程序。
（10）汇编语言　（　）J. 保存各逻辑段起始地址的寄存器，PC 机有四个：CS、DS、SS、ES。
（11）机器语言　（　）K. 控制操作的标志，PC 机有三位：DF、IF、TF。
（12）汇编程序　（　）L. 记录指令操作结果的标志，PC 机有六位：OF、SF、ZF、AF、CF、PF。
（13）连接程序　（　）M. 分析、控制并执行指令的部件，由 ALU 和寄存器组组成。
（14）指令　（　）N. 由汇编程序在汇编过程中执行的指令。
（15）伪指令　（　）O. 告诉 CPU 要执行的操作，在程序运行时执行。

【解答】（1）M　（2）C　（3）D　（4）B　（5）A
（6）L　（7）K　（8）J　（9）F　（10）H
（11）G　（12）E　（13）I　（14）O　（15）N

14. **简述汇编语言、机器语言与高级语言的区别。**

【解答】用二进制代码编写程序的语言称为机器语言，它不需要进行转换便可直接在机器上执行。但机器语言难于理解，不易记忆。

用助记符、数据项等符号书写的、其主要操作与机器指令基本上一一对应的，并遵循一定语法规则的计算机语言称为汇编语言。它与机器语言相比，需要经过汇编程序汇编产生目标代码，经过连接后才能产生可执行文件。但在表达形式上比机器语言更易于理解、编程和阅读。

用定义符和数据项等内容编写程序的语言称为高级语言。它是一种接近于自然语言的语言，它与算法有关而与机器无关。它需要经过编译程序编译转换成目标代码。相对于机器语言、汇编语言而言，转换后的目标代码所占用的内存空间大，执行程序的速度慢，但较容易编写程序，也易于阅读和理解。

第 2 章　8086/8088 的寻址方式和指令系统

微处理器指令系统的寻址方式的多寡是说明寻找操作数的灵活程度，指令系统是表现处理器解决实际问题能力的强弱，它们是汇编语言程序设计的基础。熟练掌握寻址方式和指令系统中的各种常用指令是学好本课程的关键。

通过本章的学习，要求深刻理解寻址方式、指令和指令系统的基本概念；熟练掌握各种寻址方式的含义和书写格式；深刻理解每条指令的功能和操作数的形成；灵活运用寻址方式寻找操作数和转移地址；根据运算结果，分析对 PSW 各位的影响；并能使用各种常用指令分析和编写简单的程序段。

【学习重点】

1. 操作数和转移地址的寻址方式。
2. 数据传送、算术运算、逻辑运算和移位、串操作、控制转移等指令的基本功能。

【学习难点】

串操作指令和控制转移指令的基本功能。

2.2.1　知识体系结构

在本章中，所讲述的知识体系结构如图 2.1 所示。

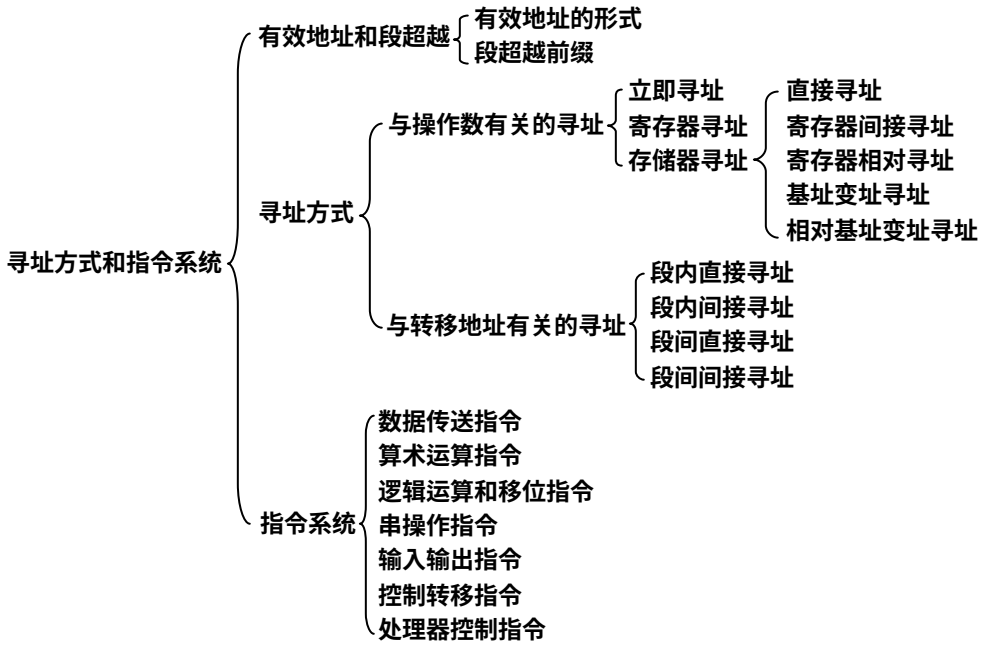


图 2.1　8086/8088 **寻址方式和指令系统知识体系结构**

2.2.2　知识点和考核要求

1. **寻址方式的定义，要求达到“识记”的层次.。**

   （1**）寻址方式的含义和实质。**

   （2**）指令中的寻址操作数可分为三种：寄存器操作数、立即数、存储器操作数（其中包括** 5 **种寻址方式）。**

2. **与数据有关的寻址方式，应达到“综合应用”的层次。**

   （1**）**8086 CPU **中与数据有关的七种寻址方式的名称和含义。**

   （2**）各种寻址方式的操作数书写格式，各种寻址方式的存储器操作数的有效地址的形成方法和书写格式。**

   （3**）分析指令中的各种寻址方式的操作数据的出处和去处，根据要求在指令中写出各种寻址方式的操作数。**

3. **与转移地址有关的寻址方式，应达到“简单应用”的层次。**

   （1**）**8086 **的指令系统中与转移地址有关的四种寻址方式的名称和含义，各种寻址方式转移地址的书写格式和转移范围的书写方式。**

   （2**）各种寻址方式中转移地址的组成和形成方法，段内间接和段间间接寻址方式中，存放转移地址的存储单元的有效地址的形成方法和转移地址各成分的存放次序。**

4. 指令和指令系统的定义，应达到“识记”的层次。

（1）什么是指令系统，熟悉指令系统中有哪些最常用的指令类型。
（2）指令的定义和指令的组成，操作码和操作数在指令中的作用。

5. 数据传送指令，应达到“综合应用”的层次。

（1）数据传送指令的助记符，源操作数和目的操作数的寻址方式、书写格式和书写顺序的规定，指令执行的操作，注意两个操作数寻址方式之间不允许的搭配关系。
（2）堆栈操作指令的助记符，操作数的寻址方式的规定、字长和书写格式，入栈/出栈操作过程：数据入栈、出栈和堆栈指针的变化情况。
（3）互换指令助记符，两个操作数的寻址方式和书写格式的规定，指令执行的操作。
（4）三种地址传送指令的助记符，源操作数和目的操作数的寻址方式、书写格式的规定，指令的执行操作。

6. 算术运算指令，应达到“综合应用”的层次。

（1）加法、带进位加法、减法、带借位减法、比较等指令的助记符；被加数或被减数（也是和或差）、加数或减数的寻址方式和书写格式的规定；指令执行的操作；两个操作数寻址方式之间不允许的搭配关系；比较指令和减法指令操作上的异同点；算述运算类指令操作结果影响标志位的情况。
（2）增 1、减 1 和求补指令的助记符；操作数的寻址方式和书写格式的规定；指令执行的操作。此类指令操作结果影响标志位的情况。
（3）无符号数和有符号数乘、除运算指令的助记符；被乘数、乘数、被除数、除数的寻址方式和书写格式的规定；乘积、商、余数所在寄存器的规定；指令执行的操作。
（4）压缩 BCD 码、非压缩 BCD 码的定义和表示法；带符号的压缩的 BCD 码的补码表示法；十进制调整的由来和调整的方法。
（5）压缩 BCD 码、非压缩 BCD 码加减调整指令助记符；记住操作数的隐含规定及调整的结果情况，记住使用此类指令的前提；操作结果影响标志位的情况。

7. 逻辑运算指令，应达到“综合应用”的层次。

（1）逻辑运算指令的助记符，两个操作数的寻址方式和书写格式的规定，两个操作数寻址方式之间不允许的搭配关系，指令执行的操作，操作结果影响标志位的情况。
（2）测试指令和逻辑与指令在操作上的异同点。
（3）逻辑非指令单操作数的寻址方式和书写格式的规定。
（4）移位指令的助记符，参与移位的操作数的寻址方式和书写格式的规定，表明移位次数的操作数的书写的规定，指令执行的操作，移位指令操作结果影响标志位的情况。

8. 串处理指令，应达到“简单应用”的层次。

（1）串处理指令的特性；连续执行时，与重复前缀 REP、REPE/REPZ、REPNE/REPNZ 配合使用的规定。

（2）五种串处理指令的助记符，源操作数和目的操作数的寻址方式、书写格式的规定，指令执行的操作，指令停止执行的条件和判别的方法。

9. 输入输出指令，应达到“理解”的层次。

输入输出指令的助记符，操作数的寻址方式和书写格式的规定，指令执行的操作。

10. 控制转移指令，应达到“综合应用”的层次。

（1）无条件转移指令的助记符，有关操作数（即转移地址）寻址方式的考核要求与 2.2.2 节中的知识点 3 相同，掌握指令执行的操作。

（2）条件转移指令和循环指令的助记符；弄清条件所属标志和标志的状态表示；操作数的寻址方式和书写格式的规定；指令执行的流程。

（3）子程序调用和返回指令的助记符，指令执行的操作；子程序调用指令的操作数的寻址方式，除子程序无近程调用以外，考核要求与本章 2.2.2 节中的知识点 3 相同。

11. 处理器控制指令，应达到“简单应用”的层次。

处理器控制指令的助记符和指令执行的操作。

2.3 例题分析

例 1　设 A 为字变量，指令“MOV　AX，A”的源操作数寻址方式是（　　）。

A. 立即　　B. 直接　　C. 寄存器　　D. 寄存器相对

解：因为 A 是字变量，在汇编的过程中，它将 A 的偏移地址替换了 A。所以在本题中应选择 B。

例 2　下列指令中出错的是（　　）。

A. MOV　BUF，S　　B. ADD　AX，[BX+SI]

C. JMP　WPRD PRT[BX]　　D. AND　AX，08H

解：因为选项 A 中的双操作数都是变量，如果两个变量都是内存中的数据，在 8086、8088 的指令系统中，这种情况是不允许的。如果两个变量中有立即数（即 S 被定义为“EQU”或“＝”的立即数），此时双操作数都没有指出是字节还是字传送，使得指令的操作存在二义性。所以也会出现汇编错误。本题中应选 A。

例 3　指令 MOV　BX，A[SI]的源操作数属于_____寻址，指令 JMP DWORD PRT[BX]属于_____寻址。

解：指令 MOV　BX，A[SI]**的源操作数为** A[SI]，**它是以** SI **中的内容加** A **的位移量，所以它应属于寄存器相对寻址。指令** JMP　DWORD　PRT [BX]**的寻址方式中，是要将内存中的双字的低** 16 **位送** IP，**高** 16 **位送** CS。**这条转移指令不但改变了** IP **的值，而且也改变了** CS **的值，所以属于段间间接转移。**

例 4　**两个无符号数进行比较时，可以使用_____标志位来判断大小，在编程中使用“**_____ L **”指令实现当高于时转移到** L。

解：当两个无符号数进行比较时，是将目的操作数减去源操作数，如果目的操作数大于源操作数，也就会产生借位，其结果一定为正数。在第一个填空中，显然可以用进位位 CF **或符号位** SF **表示大小，但题目中给出的是无符号数的运算，所以选填进位位** CF **更合适。同理，在第二个填空中应用** JA **或** JNB **都可以。**

例 5　**执行相对寻址的分支转移指令时，如果操作码放在** 00A8H **处，位移量为** 80H，**那么转移后取下一条指令的地址偏移地址为（　　）。**

解：因为相对寻址的转移指令占用两个字节 00A8H **和** 00A9H **单元，当取出当前指令后，**IP **的值将自动加** 2，**指向下一条指令的地址，此时** IP=00AAH。**如果要执行转移，则** IP **的值为当前** IP **的值＋位移量（即计算转移的公式：目的地址** IP=**当前指令地址** IP+2 **＋位移量）。由于位移量是带符号数的数据，此时计算偏移量需将** 8 **位位移量扩展成** 16 **位的数据，因为此时的位移量是负数，将** 80H **进行符号扩展后变为** FF80H，**即** IP←00A8H+2+FF80H。**所以应填** 002AH。

例 6　**判断下列指令是否正确，如果是错误的，请说明原因。**

1. ADD　　AX，BL
2. MOV　　[BX]，5
3. MOV　　[DI]，[SI]
4. CMP　　5，[BX]
5. MOV　　CS，AX
6. SUB　　AX，[DI+SI]
7. MOV　　[DX]，AX
8. LEA　　BX，P

解：第 1 **条指令是错误的指令，因为两个操作数的类型不一致。**

第 2 **条指令是错误的指令，因为操作数的类型不明确（是字还是字节？）。**

第 3 **条指令是错误的指令，因为两个操作数都是内存中的数据，这种情况是不允许的。**

第 4 **条指令是错误的指令，因为立即数不能作为目标地址，同时还存在着操作数的类型不明确的错误。**

第 5 **条指令是错误的指令，因为** CS **不能由用户任意修改。**

第 6 **条指令是错误的指令，因为** DI **和** SI **不能同时使用一种寻址方式。**

第 7 **条指令是错误的指令，因为** DX **寄存器不能作为寄存器间接寻址，用作寄存器间接寻址的寄存器只有** BX、BP、SI **和** DI。

第 8 条指令是正确的。

例 7　根据指令的原意修正下列各条指令的错误。

```
1. INC     [BX]
2. MOV     AX，[DX+SI]
3. MOV     BX，OFFSET[SI]
4. CMP     AL，1000H
5. IN      AX，300H
```

解：第 1 条指令的操作数没有指出是字还是字节的操作，指令存在着二义性，如果是字操作，则应改为 INC WORD PTR[BX]；如果是字节操作，则应改为 INC BYTE PTR[BX]。

第 2 条指令中的源操作数中使用 DX 是不正确的，因为用来作为寄存器间接寻址的寄存器只有 BX、BP、SI、DI。按题意应改为：

```
ADD   SI，DX        或      ADD   DI，SI        或      MOV   BX，DX
MOV   AX，[SI]              MOV   AX，[DI]              MOV   AX，[BI+SI]
```

第 3 条指令的源操作数中 OFFSET 后面只能跟标号或者是变量，不能是表达式。所以本条指令应改为：LEA　BX，[SI]。

第 4 条指令的 8 位寄存器 AL 中的内容与 16 位数据 1000H 比较，操作数的类型不一致。如果是无符号数的比较，则应将 AL 中的 8 位数据作为 16 位数据的低 8 位，并在高 8 位中补 0。（即 AH 送全 0 后，使 AX 中存放的数与 AL 中的数不变），所以应改为：

```
MOV     AH，0
CMP     AX，1000H
```

如果是有符号数的比较，则应将 AL 中的数据进行符号扩展，将 AL 中的 8 位数据扩展成 16 位数据后再比较。所以应改为：

```
CBW
CMP     AX，1000H
```

第 5 条输入指令中的 I/O 端口地址 300H 超出了 0～255 的范围，不能采用直接寻址方式，需用 DX 寄存器作为端口地址的寄存器，所以应改为：

```
MOV     DX，300H
IN      AX，DX
```

例 8　设 X、Y、Z 变量均为 16 位有符号数，分析下面程序段完成的功能。

```
MOV    AX，X
IMUL   Y
MOV    CX，AX
MOV    BX，DX
MOV    AX，Z
```

```
CWD
ADD    CX，AX
ADC    BX，DX
SUB    CX，540
SBB    BX，0
MOV    AX，V
CWD
SUB    AX，CX
SBB    DX，BX
IDIV   X
```

解：前 4 条指令是将 X*Y 的结果送 BX：CX（BX 存放高 16 位，CX 存放低 16 位）。第 5、6 条指令是将 Z 中的 16 位数据扩展成 32 位的数据存放在 DX：AX 中，准备和 X*Y 的积相加。第 7、8 条指令是将 X*Y+Z 的结果送 BX：CX。第 9、10 条指令是在原有计算结果的基础上减去 540，并将结果送 BX：CX，完成 BX：CX←X*Y+Z-540 的操作。第 11、12 条指令是将 V 扩展成 32 位的数据存放在 DX：AX 中。第 13、14 条指令是将 DX：AX 中的 32 位数据减去 BX：CX 中的数据，完成 DX：AX←V-（X*Y+Z-540）的操作。最后将 DX：AX 中的 32 位数据除以 16 位数据 X，所以本段程序的功能为：

（V-（X*Y+Z-540））/X

其中商存放在 AX 中，余数存放在 DX 中。

例 9　分析下列程序段完成的功能。

```
MOV    CL，04
SHL    DX，CL
MOV    BL，AH
SHL    AX，CL
SHR    BL，CL
OR     DL，BL
```

解：前 2 条指令是将 DX 中的数据逻辑左移 4 位，低 4 位补 0。第 3 条指令是复制原来 AH 中的数据到 BL 中。第 4 条指令是将 AX 中的数据逻辑左移 4 位，低 4 位补 0。第 5、6 条指令是将原来 AH 中的数据（现在存放在 BL 中）逻辑右移 4 位，使原来 AH 中的最高 4 位数据移到 AH 的低 4 位，最后将此 4 位数据送到 DX 中空出的 4 位（低 4 位）。由此可以看出，DX、AX 寄存器中原来的数据都左移了 4 位，同时将 AX 寄存器中原来的高 4 位数据送到了 DX 寄存器中的低 4 位，所以此程序段完成的功能是将 DX：AX 中的 32 位数据逻辑左移 4 位。

例 10　设 B1=4234H，B2=3412H，现有下列程序段：

```
MOV   AX，B1
MOV   BX，B2
MOV   CH，AH
SUB   AL，BL
```

```
DAS
MOV   B3，AL
MOV   AL，CH
SBB   AL，BH
DAS
MOV   B3+1，AL
```

请回答：（1）**该程序段完成什么功能？**

（2）**执行该程序段后，**B3=_____。

解：该程序段的前 2 **条指令是将** B1 **和** B2 **的内容分别送** AX **和** BX **寄存器，第** 3 **条指令是将** B1 **的高** 8 **位数据送** CH **寄存器保存。第** 4～6 **条指令是将** B1 **的低** 8 **位数据和** B2 **的低** 8 **位数据的两个压缩型的** BCD **码相减，即** 34H-12H=22H。**由于相减后既没有产生借位（**CF=0**），也没有产生辅助（**AF=0**）借位，所以** DAS **调整后还是为** 22H**，并将此结果送给** B3。**后** 4 **条指令是将** B1 **的高** 8 **位数据和** B2 **的高** 8 **位数据的两个压缩型的** BCD **码带借位相减，即** 42H-34H=0EH。**由于在减法过程中产生了辅助借位，在进行调整时应进行减** 6 **修正。所以** 0EH-6=08H **送** B3+1。**由此可得，本段程序完成的功能是** B1-B2→B3。**执行完该段程序后** B3 **的两个存储单元中的内容依次为** 22H、08H。

例 11　**现有程序段如下：**

```
    MOV   CX，16
    MOV   BX，0
    MOV   DX，1
L1：MOV   AX，2AB0H
    TEST  AX，DX
    JZ    L2
    INC   BX
L2：SHL   DX，1
    LOOP  L1
    MOV   MEM，BX
```

请回答：（1）**该程序段完成什么功能？**

（2）MEM **中的内容是**_____。

解：该程序段中，CX **寄存器作为循环计数值计数器，共计循环** 16 **次。**BX **寄存器作为计数** AX **寄存器中为** 1 **的位数计数器。**DX **寄存器作为测试** AX **寄存器中有哪一位为** 1 **的逻辑尺，首先将** DX **寄存器中的最低位设置成** 1，**其他位设置成** 0。**当** AX **中的内寄与** DX **中的内容相与时，此时测试的是** AX **中对应** DX **为** 1 **的位是否为** 1。**如果为** 1，**则相与后的结果一定不为** 0，**当执行** INC BX **指令后，将** BX **中的内容加** 1；**如果** AX **中的最低位不为** 0，**则相与后的结果一定为** 0。**此时，程序转移到** L2，DX **中为** 1 **的位左移一位，移至次低位，则转移到** L1 **继续依次重复** 16 **次，就可统计出** AX **中为** 1 **的个数。**

该段程序的功能是统计 AX **中为** 1 **的个数，统计的个数放在** BX **中。**

该段程序执行完后，存储单元 MEM **中的内容是** 6。

2.4 练习题与参考答案

2.4.1 单项选择题

1. 设 BX=2000H，SI=3000H，指令 MOV AX，[BX+SI+8]的源操作数的有效地址为（　　）。

A. 5000H　　B. 5008H　　C. 23008H　　D. 32008H

2. 设 DS=1000H，ES=2000H，BX=3000H，指令 ADD AL，[BX]的源操作数的物理地址为（　　）。

A. 13000H　　B. 23000H　　C. 33000H　　D. 3000H

3. 设 DS=2000H，ES=3000H，SI=200H，指令 MOV ES：[SI]，AL 的目的操作数的物理地址为（　　）。

A. 20200H　　B. 30200H　　C. 50200H　　D. 200H

4. 指令 MOV MEM[BX]，AX 中的 MEM 是（　　）。

A. 原码　　B. 反码　　C. 补码　　D. 移码

5. 用来作为寄存器间接寻址的寄存器有（　　）个。

A. 8　　B. 6　　C. 5　　D. 4

6. 指令 MOV [BX+SI]，AL 中的目的操作数使用（　　）段寄存器。

A. CS　　B. DS　　C. SS　　D. ES

7. 指令 MOV BX，[BP+5]中的源操作数使用（　　）段寄存器。

A. CS　　B. DS　　C. SS　　D. ES

8. 段内间接寻址只改变（　　）中的内容。

A. CS　　B. IP　　C. CS 和 IP　　D. PSW

9. 段间间接寻址只改变（　　）中的内容。

A. CS　　B. IP　　C. CS 和 IP　　D. PSW

10. 下述指令中不改变 PSW 的指令是（　　）。

A. MOV AX，BX　　B. AND AL，0FH
C. SHR BX，CL　　D. ADD AL，BL

11. 下述指令中不影响 CF 的指令是（　　）。

A. SHL　AL，1　　B. INC　CX
C. ADD　[BX]，AL　　D. SUB　AX，BX

12. 两个 8 位二进制数的整数补码 9CH 和 7AH 进行相加运算后，会产生（　　）。

A. 无溢出且无进位　　B. 无溢出但有进位
C. 有溢出且有进位　　D. 有溢出但无进位

13. 指令 JMP　WORD　PTR [BX]属于（　　）寻址。

A. 段内直接　　B. 段内间接　　C. 段间直接　　D. 段间间接

14. 指令 MOV　AX，[BX+SI+8]的源操作数属于（　　）寻址。

A. 直接　　B. 寄存器相对
C. 基址变址　　D. 相对基址变址

15. 指令（　　）不改变 CF 的内容。

A. DEC　AL　　B. ADD　AX，CX
C. SUB　[BX]，CL　　D. SBB　AL，DL

16. 十进制数字 74 所对应的压缩型 BCD 码的形式是（　　）。

A. 74　　B. 74H　　C. 4AH　　D. 4A

17. 十进制数字 85 所对应的非压缩型 BCD 码的形式是（　　）。

A. 0085　　B. 0085H　　C. 0805　　D. 0805H

18. 设 AL=67H，执行“CMP　AL，76H”后，AL=（　　）。

A. 76H　　B. 0DFH　　C. 67H　　D. 00

19. 设 AL=65H，BL=29H，执行下列指令后，AL=（　　）。

```
ADD　AL，BL
DAA
```

A. 8EH　　B. 94　　C. 94H　　D. 8E

20. 入栈操作是（　　）位数的操作。

A. 8　　B. 16　　C. 32　　D. 任意

21. 执行 IMUL 指令时，如果乘积的高位部分不是低位部分的符号扩展，则（　　）。

A. OF=0、CF=0　　B. OF=0、CF=1
C. OF=1、CF=0　　D. OF=1、CF=1

22. 设 AX=3762H，CL=5，执行“SHR　AX，CL”后，AX=（　　）。

A. 0376H　　B. 01BBH　　C. 01BB　　D. 0376

23. 若要在 BUF 缓冲区中寻找与 AL 中不相等的数据，应使用（　　）SCASB 串操作指令。

A. REPNE　　B. REP　　C. REPE　　D. REPNZ

24. 如果“JNC　L”指令的操作码放在 0040H，转移后在 0020H 处取下一条指令的操作码，那么该条指令的位移量是（　　）。

A. 20H　　B. 1EH　　C. 0DEH　　D. 0E0H

25. 如果“JA　P”指令的操作码放在 0050H，该指令的位移量为 34H，执行完此条指令转移取下一条指令的偏移地址为（　　）。

A. 0082H　　B. 0084H　　C. 0086H　　D. 0088H

26. 若 DS=1000H，BX=2000H，(12000H)=56H，(12001H)=78H，AX=1000H，执行“ADD　AX，[BX]”指令后，AX=（　　）。

A. 8856H　　B. 6678H　　C. 8800H　　D. 6600H

27. 设 AX、BX 寄存器中存放的是有符号的二进制数据，若执行“CMP　AX，BX”指令后，“（　　）L”表示大于则转 L。

A. JZ　　B. JA　　C. JGE　　D. JG

28. 执行“DIV　BX”指令后，（　　）寄存器中存放商。

A. AL　　B. AH　　C. AX　　D. DX

29. 执行“（　　）AX，BX”指令不改变 AX 寄存器中的内容。

A. CMP　　B. ADD　　C. XOR　　D. OR

30. 若 AX=1000H，执行“NEG　AX”指令后，AX=（　　）。

A. 1000H　　B. 0E000H　　C. 0F000H　　D. 1001H

A 单项选择题参考答案

1. B	2. A	3. B	4. C	5. D	6. B
7. C	8. B	9. C	10. A	11. B	12. B
13. B	14. D	15. A	16. B	17. D	18. C
19. C	20. B	21. D	22. B	23. C	24. C
25. C	26. A	27. D	28. C	29. A	30. C

2.4.2 多项选择题

1. **在下列寻址方式中，用来访问内存的寻址方式有（　　）。**

A. **寄存器寻址**　　B. **寄存器间接寻址**
C. **寄存器相对寻址**　　D. **直接寻址**

2. **用来作为寄存器间接寻址的寄存器有（　　）。**

A. AX　　B. BX　　C. BP　　D. CX

3. **在下列指令中，源操作数使用** DS **段寄存器进行寄存器相对寻址的有（　　）。**

A. MOV　AX，[DI+4]　　B. MOV　AX，ES：[SI+8]
C. MOV　AX，[BP+4]　　D. MOV　AX，[BX+4]

4. **在下列指令中，源操作数的寻址方式是错误的有（　　）。**

A. MOV　AX，[DI+BX]　　B. MOV　AX，[SI+DI]
C. MOV　AX，[BP+BX]　　D. MOV　AX，[DX]

5. **在下列指令中，属于段内转移指令的有（　　）。**

A. JMP　SHORT　A　　B. JMP　[BX]
C. JMP　DWORD PTR [BX]　　D. JMP　NEAR　PTR [BX+SI]

6. **在下列指令中，错误的指令有（　　）。**

A. SUB　5，AL　　B. ADD　AL，BX
C. INC　[BX]　　D. SHR　AX，6

7. **执行“**CMP　AX，8003H**”指令后，当** AX **中的无符号数高于** 8003H **时，下列指令中有效的转移指令有（　　）。**

A. JNB　L　　B. JA　L　　C. JG　L　　D. JNL　L

8. **可与串操作指令“**CMPSW**”指令配合使用的重复前缀有（　　）。**

A. REP　　B. REPZ　　C. REPNZ　　D. REPE

9. **在下列的输入输出指令中，正确的指令有（　　）。**

A. IN　AX，80H　　B. OUT　DX，AX
C. IN　AL，340　　D. OUT　DX，AL

10. **在下列的指令中，错误的指令有（　　）。**

A. PUSH　AL　　B. MOV　AL，BX
C. PUSH　1000H　　D. CALL　AX

11. 可以用下列指令代替“ LOOP　L ”指令的有（　　）。

A.
```
DEC　CX
JNZ　L
```
B.
```
DEC　CX
JNC　L
```
C.
```
DEC　CX
CMP　CX，0
JE　L
```
D.
```
DEC　CX
JE　L
```

多项选择题参考答案

1. BCD	2. BC	3. AD	4. BCD	5. ABD
6. ABCD	7. AB	8. BCD	9. ABD	10. ABCD
11. AC				

2.4.3　填空题

1. 在一条指令中，立即数只能作_____操作数。
答：源

2. 8086/8088 CPU 形成的内存物理地址有_____位。
答：20

3. 指令“ MOV　AX，[BX+SI] ”的源操作数在内存的_____段。
答：DS（数据）

4. 指令“ MOV　BX，[BP+DI] ”的源操作数在内存的_____段。
答：SS（堆栈）

5. 指令“ MOV　AX，ES：[SI] ”的源操作数在内存的_____段。
答：ES（附加）

6. 入栈指令使用的是_____段。
答：SS（堆栈）

7. 指令“ ADD　[BX+SI]，AL ”的目的操作数是_____寻址方式。
答：基址加变址

8. 指令“ SUB　BX，A[SI] ”的源操作数是_____寻址方式。
答：寄存器相对

9. 指令“ JMP　DWORD　PTR　[BX] ”属于_____寻址方式。
答：段间间接

10. 可作为寄存器间接寻址的寄存器有_____个。

答：4

11. 堆栈是从_____地址向_____方向生长的。其操作遵循_____的操作原则。

答：高　低　后进先出（先进后出）

12. 在进行出栈操作时应该先输出数据送_____，然后_____ 。

答：目标地址　SP+2

13. 在进行多精度加法操作时，一般使用_____指令。

答：ADC（带进位加）

14. 通用的数据传送指令不影响_____。

答：标志位（PSW）

15. “ INC　AL ”指令不影响标志位的_____位。

答：CF

16. 若 AL=11H，执行“ NEG　AL ”后，AL=_____。

答：0EFH

17. JMP 指令的执行_____PSW 寄存器中的各位。

答：不影响

18. 两个无符号数进行比较时，可以根据_____标志位来判断大小。在编程中可使用_____指令来实现。

答：CF（SF）　JA（JB、JC、JNC、JS、JNS）

19. 若执行“ DIV　BX ”，其被除数的高 16 位必须放在_____，低 16 位必须放在_____。

答：DX　AX

20. DAA 指令只对_____寄存器中的内容进行调整。

答：AL

21. 若 AL=84H，在执行 CBW 后，AL=_____。

答：0FF84H

22. 十进制数 1234 的压缩型 BCD 码为_____。

答：1234H

23. 执行 AND　AL，0FH

JNZ　L

指令，是指检测 AL 中的_____位，当为非零时转移。

答：低 4 位

24. 若移位指令的移位位数大于 1 时，其移位位数必须放在_____中。

答：CL

25. 如果 AL=85H，CL=4，执行“SAR　AL，CL”时，AL=_____，CF=_____。
答：0F8H　0

26. 在串操作指令中，SI 指向_____串，DI 指向_____串。
答：源　目的

27. 如果要查找某串中与 AL 寄存器中有相同的字符（数），则在 SCASB 中配合使用_____重复前缀。
答：REPNE（REPNZ）

28. 转移指令“JNZ　L”的转移范围（十进制）是_____。该指令的第二个字节为位移量，用_____形式表示。
答：–128～127　**补码**

29. 段内调用指令改变_____中的内容。
答：IP

30. 段间调用指令改变_____中的内容。
答：CS **和** IP

2.4.4　判断题

1. ADD　AL，BX
答：错，两个操作数的类型不一致。

2. MOV　[BX]，SI
答：对。

3. SUB　[BX]，[SI]
答：错，双操作数不能同时是内存中的数据。

4. CMP　5，AL
答：错，5 **是立即数，不能作目标地址。**

5. MOV　CX，AX
答：对。

6. LEA　BL，A
答：错，A **的偏移地址是** 16 **位的，而** BL **是** 8 **位的寄存器，两个操作数的类型不一致。**

7. SHR　AL，6
答：错，移位指令的移位位数大于 1 **时，其移位位数应存放在** CL **中。**

8. MOV　AL，[DI+SI]
答：错，DI **和** SI **不能同时在一种寻址方式中使用。**

9. MOV　BL，200H

答：错，200H **超出一个字节的范围，两个操作数的类型不一致。**

10. MOV　AX，DATA

答：对。

11. INC　[BX]

答：错，[BX]**是字节还是字操作，不明确，有二义性。**

12. CMP　[BX]，A　　；A **为变量**

答：错，双操作数不能同时是内存中的数据。

13. MOV　AX，‘ + ’

答：对。

14. MOV　AX，[BP+BX]

答：错，BP **和** BX **不能同时使用。**

15. ADD　AX，[CX]

答：错，CX **不能作寄存器间接寻址。**

16. OUT　400，AL

答：错，OUT **指令的端口地址超过** 255 **时，必使用** DX **作间接寻址。**

17. JNC　P

答：对。

18. MOV　BX，OFFSET A

答：对。

19. LEA　DI，A[SI]

答：对。

20. CMP　AX，5

答：对。

2.4.5　改错题

1. DEC　[SI]

答：DEC　　BYTE　PTR　[SI]　　**或**　DEC　WORD　PTR　[SI]

2. CMP　AL，BX　　；**无符号数比较**

答：MOV　　AH，0

　　CMP　　AX，BX

3. MOV　[DX]，AL

答：MOV　　SI，DX　　**或**　　MOV　DI，DX　　**或**　　MOV　BX，DX
　　MOV　　[SI]，AL　　　　MOV　[DI]，AL　　　　MOV　[BX]，AL

4. MOV　AL，300H

答：MOV　　AX，300H

5. MOV　[BX]，[DI]　　**；字节操作**

答：MOV　　AL，[DI]
　　MOV　　[BX]，AL

6. ADD　AL，[CX+SI]

答：ADD　　SI，CX
　　ADD　　AL，[SI]

7. ADD　AL，BX　　**；有符号数相加**

答：CBW
　　ADD　　AX，BX

8. PUSH　AL

答：PUSH　　AX

9. CMP　[BX]，5　　**；字节比较**

答：CMP　BYTE　PTR　[BX]，5　　**或**　　CMP　[BX]，BYTE　PTR　5

10. MOV　DS，1234H

答：MOV　　AX，1234H
　　MOV　　DS，AX

2.4.6　简答题

1. **设** BX=1000H，SI=2000H，**位移量** D=3000H，**请指出下列各种寻址方式的有效地址是什么？**

（1）**使用** D **的直接寻址**
（2）**使用** BX **寄存器的间接寻址**
（3）**使用** BX **寄存器的相对寻址**
（4）**基址变址寻址**
（5）**相对基址变址寻址**

【解答】（1）3000H
（2）1000H
（3）4000H
（4）3000H
（5）6000H

2. **请指出下列各条指令的源操作数的寻址方式是什么？**

（1）MOV　AX，BUF
（2）CMP　AL，5
（3）ADD　[BX+5]，AX
（4）SUB　AX，[BX+SI]
（5）ADC　AH，A[SI]
（6）MOV　BX，[SI]
（7）AND　CX，B[BP+SI]
（8）JMP　WORD　PTR　[BX]
（9）JMP　P
（10）JMP　FAR　PTR　P

【解答】（1）**直接**
（2）**立即**
（3）**寄存器**
（4）**基址加变址**
（5）**寄存器相对**
（6）**寄存器间接**
（7）**相对基址变址**
（8）**段内间接**
（9）**段内直接**
（10）**段间直接**

3. **设** DS=2000H，BX=0100H，SI=0002H，（20100H）=12H，（20101H）=34H，（20102H）=56H，（20103H）=78H，（21200H）=2AH，（21201H）=4CH，（21202H）=B7H，（21203H）=65H，**试说明下列各条指令执行完后** AX **寄存器中的内容是多少？**

（1）MOV　AX，1200H
（2）MOV　AX，BX
（3）MOV　AX，[1200H]
（4）MOV　AX，[BX]
（5）MOV　AX，[BX+1100H]
（6）MOV　AX，[BX+SI]
（7）MOV　AX，[BX+SI+1100H]

【解答】（1）1200H
（2）0100H
（3）4C2AH
（4）3412H
（5）4C2AH
（6）7856H

（7）65B7H

4. **按下列各小题的要求写出相应的一条汇编语言指令。**

（1）把 BX 寄存器和 DX 寄存器的内容相加，结果存入 DX 寄存器中。
（2）以 BX 和 SI 寄存器作基址变址寻址方式，把该单元中的一个字传送到 AX。
（3）以 SI 和位移量 20H 作寄存器相对寻址，将该单元中的内容与 CX 寄存器中的内容相加，结果存入 CX 寄存器中。
（4）清除 AX 寄存器的内容，同时清除 CF 标志位。
（5）将字单元 NUM 与 0B6H 进行比较。

【解答】（1）ADD　　DX，BX
（2）MOV　　AX，[BX+SI]
（3）ADD　　CX，[SI+20H]
（4）XOR　　AX，AX
（5）CMP　　WORD PTR NUM，0B6H

5. **按下列各小题的要求使用相应的几条指令完成其操作。**

（1）将偏移量为 200H 的存储单元中的数与 300H 相加，结果存入 AX 寄存器中
（2）比较 AX 寄存器中与 BX 寄存器中的 16 位的有符号数，当 AX 中的内容大于 BX 中的内容时转移到 L。
（3）将 BUF1 和 BUF2 中的 16 位的数据交换。
（4）测试 BUF 字缓冲区中第 15 位，如果为 1 则转移到 P。
（5）将 BUF1 和 BUF2 的字相加，结果送 S 缓冲区。

【解答】（1）MOV　AX，[200H]
　　ADD　AX，300H
（2）CMP　AX，BX
　　JG　L
（3）MOV　AX，BUF1
　　XCHG　AX，BUF2
　　MOV　BUF1，AX
（4）TEST　WORD PTR BUF，8000H
　　JNZ　P
（5）MOV　AX，BUF1
　　ADD　AX，BUF2
　　MOV　S，AX

6. **设 BX=8234H，请说明下列两条指令的区别，执行下列各指令后 BX 中的内容是什么？**

SHR　BX，1

SAR　BX，1

【解答】SHR 是逻辑右移指令，移位后空出的位补 0，而 SAR 是算术右移指令，最高位保持不变，其他位右移一位。执行“ SHR　BX，1 ”指令后 BX=411AH；执行“ SAR　BX，1 ”指令后 BX=0C11AH。

7. 分别说明下列每组指令中的两条指令的区别。

（1）MOV　BX，BUF　　LEA　BX，BUF
（2）OR　BL，0FH　　AND　BL，0FH
（3）JMP　SHORT　L　　JMP　L
（4）MOV　AX，BX　　MOV　AX，[BX]
（5）MOV　AX，[BX+DI]　　MOV　AX，[BP+DI]

【解答】（1）“ MOV　BX，BUF ”中的 BUF 是直接寻址，它是将 BUF 存储单元中的内容传送给 BX。而“ LEA　BX，BUF ”指令是将 BUF 的偏移地址送 BX。

（2）OR 是逻辑或指令，而 AND 是逻辑与指令。

（3）“ JMP　SHORT　L ”是短转移，产生的目标代码的位移量是 8 位的，而“ JMP　L ”产生的位移量是 16 位的。

（4）“ MOV　AX，BX ”的源操作数是 BX 中的内容，即寄存器寻址。而“ MOV　AX，[BX] ”中的源操作数是 BX 所指向的内存单元中的内容，即寄存器间接寻址。

（5）“ MOV　AX，[BX+DI] ”指令中的源操作数使用 BX 寄存器作基址寄存器，所以段寄存器应该使用 DS，而“ MOV　AX，[BP+DI] ”指令的源操作数使用 BP 寄存器作基址寄存器，所以段寄存器应使用 SS。

8. 说明下列各指令的源操作数和目的操作数的存储地方。

（1）MOV　[2000H]，AX
（2）ADD　AX，[BX+5]
（3）LEA　SI，BUF
（4）DAA
（5）MUL　BL

【解答】（1）源操作数存放在 AX 寄存器中，结果存放在地址为 2000H 的单元中。

（2）源操作数存放在内存中，结果存放在 AX 中。

（3）源操作数在指令中，即 BUF 的偏移地址，结果存放在 SI 寄存器中。

（4）源操作数和结果都存放在 AL 中。

（5）源操作数存放在 BL 中，结果存放在 AX 中。

9. 指出下列无条件转移指令的转移目标地址是什么，存放在何处。

（1）JMP　BX

（2）JMP　　WORD PTR [BX]
（3）JMP　　DWORD PTR [BX]
（4）JMP　　P
（5）JMP　　SHORT　R

【解答】（1）BX 中的内容作为目标地址的偏移地址。
（2）BX 所指字单元中的内容作目标地址的偏移地址。
（3）BX 所指字单元中的内容作目标地址的偏移地址，下一个字单元中的内容作段地址。
（4）P 的地址。
（5）R 的地址。

10. **说明** MOVSB **和** CMPSB **各能使用哪些重复前缀？**

【解答】MOVSB **只能使用** REP **重复前缀，**CMPSB **能使用** REPZ，REPNE，REPE **和** REPNZ **重复前缀。**

2.4.7　程序分析题

1. **现有程序段如下：**

```
MOV       AX，1234H
MOV       BX，60H
ADD       AX，BX
```

请回答：（1）**该程序段完成的功能是什么？**
（2）**程序段执行完后** AX=_____。

【解答】（1）**两数** 1234H **和** 60H **相加，结果存放在** AX **中。**
（2）AX　1294H

2. **现有程序段如下：**

```
MOV       AX，0603H
MOV       BL，8
AAD
DIV       BL
```

请回答：（1）**该程序段完成的功能是什么？**
（2）**程序段执行完后** AX=_____。

【解答】（1）**实现两个非压缩型** BCD **码的除法。**
（2）AX　0707H

3. **设** AX=0D023H，BX=9FD0H，**试分析执行完如下程序段后程序转向何处？**

```
ADD       AX，BX
JNO       L1
```

```
JNC     L2
SUB     AX，BX
JNC     L3
JNO     L4
JMP     L5
```

【解答】L5

4. **现有程序段如下：**

```
MOV     AX，X
MOV     DX，X+2
ADD     AX，Y
ADC     DX，Y+2
ADD     AX，36
ADC     DX，0
SUB     AX，Z
SBB     DX，Z+2
MOV     W，AX
MOV     W+2，DX
```

请回答：（1）**该程序段完成的功能是什么？**
（2）**该程序的操作数是何类型？**
（3）**结果存放在何处？**

【解答】（1）**计算** X+Y+36-Z
（2）**字**
（3）W

5. **设** A **变量中存放的字节数据为** 64H、52H，B **变量中存放的字节数据为** 12H、46H，**现有程序段如下：**

```
MOV     AL，A
SUB     AL，B
DAS
MOV     C，AL
MOV     AL，A+1
SBB     AL，B+1
DAS
MOV     C+1，AL
```

请回答：（1）**该程序段完成的功能是什么？**
（2）**最后结果是什么？**

【解答】（1）**实现两个压缩型** BCD **码的减法，即** A-B，**结果送** C。
（2）0652H

6. **现有程序段如下：**

```
MOV     BL，AL
MOV     CL，4
SHR     BL，CL
MOV     A，BL
AND     AL，0FH
MOV     B，AL
```

请回答：(1) **该程序段完成的功能是什么？**

(2) **如果** AL **的初值为** 56H，**则** A=_____，B=_____。

【解答】(1) **将** AL **中的** 8 **位数据分高** 4 **位和低** 4 **位分别存放在** A **和** B **中。**

(2) A=05H，B=06H

7. **现有程序段如下：**

```
MOV     AL，8
MOV     BL，7
MUL     BL
AAM
ADD     AX，3030H
XCHG    AH，AL
MOV     BUF，AX
```

请回答：(1) **该程序段完成的功能是什么？**

(2) BUF **两个单元中的值分别为**_____。

【解答】(1) **将两个数相乘的结果转换成** ASCII **码存放在** BUF **中。**

(2) 35H，36H

8. **现有程序段如下：**

```
MOV     AX，M
MOV     DX，N
SHR     DX，1
RCR     AX，1
```

请回答：(1) **该程序段完成的功能是什么？**

(2) **若** M=1234H，N=5678H，**程序运行后** DX=_____ ，AX=_____ 。

【解答】(1) **将** N:M **中的** 32 **位右移一位。**

(2) DX=091AH，AX=2B3CH

9. **现有程序段如下：**

```
XOR     AX，AX
MOV     AX，6342H
MOV     CX，0404H
```

```
ROL     AH，CL
XCHG    CH，CL
ROR     AL，CL
```

请回答：（1）该程序段执行后 AX=_____。

（2）CF= _____。

【解答】（1）AX=3624H

（2）CF=0

10. **现有程序段如下：**

```
        MOV     AX，1
        MOV     BX，2
        MOV     CX，4
        MOV     DX，3
L：     INC     AX
        ADD     BX，AX
        SHR     DX，1
        LOOPNZ  L
```

请回答：（1）该程序段的循环次数是多少？

（2）该程序段执行完后 AX=_____，BX=_____，CX=_____，DX=_____。

【解答】（1）2

（2）AX=3，BX=7，CX=4，DX=0

11. **现有程序段如下：**

```
        MOV     CX，16
        MOV     BX，0
        MOV     DX，1
L：     MOV     AX，9AB8H
        AND     AX，DX
        JZ      N
        INC     BX
N：     SHL     DX，1
        LOOP    L
        MOV     M，BX
```

请回答：（1）该程序段完成的功能是什么？

（2）该程序段执行完后（M**）=**_____。

【解答】（1）统计 9AB8H **中为** 1 **的位数。**

（2）M=8

12. **现有程序段如下：**

```
CLD
MOV     SI，OFFSET  BUF1
MOV     DI，OFFSET  BUF2
MOV     CX，100
REP     MOVSB
```

请回答：该程序段完成的功能是什么？

【解答】将 BUF1 **中** 100 **个字节的数据传送到** BUF2 **中。**

13. **现有程序段如下：**

```
CLD
LEA     DI，BUF
MOV     AL，20H
MOV     CX，100
REPNZ   SCASB
```

请回答：（1）**该程序段完成的功能是什么？**

（2）**若** ZF=1，**表示** BUF **中_____值为** 20H **的数据。**

（3）**若** BUF **的首地址为** 0，ZF=0，**执行完该程序段后** DI=_____ **。**

【解答】（1）**查找** BUF **中为空格（**20H**）的字符。**

（2）**有**

（3）DI=100

14. **现有程序段如下：**

```
        MOV     AL，0
        MOV     BL，1
        MOV     CX，10
L：     ADD     AL，BL
        INC     BL
        LOOP    L
```

请回答：（1）**该程序段完成的功能是什么？**

（2）**该程序执行完后** AL=_____ **。**

【解答】（1）**实现** 1+2+3+…+10

（2）AL=37H（55）

15. **现有程序段如下：**

```
        MOV     AX，M
        CMP     AX，N
        JA      L
        MOV     AX，M+2
```

```
        CMP     AX，N+2
        JAZ     L
        MOV     FLG，BYTE  PTR 0
        JMP     P
L：     MOV     FLG  BYTE  PTR 1
P：     …
```

请回答：（1）**该程序段完成的功能是什么？**

（2）**若** M=1234H，N=5678H，**则** FLG=_____。

【解答】（1）**两个** 32 **位的无符号数** M **和** N **的比较，大小等于时** 1→FLG；**否则** 0→FLG。

（2）FLG=0

16. **现有程序段如下：**

```
        MOV     CX，100
        LEA     BX，BUF
        MOV     DX，0
L1：    MOV     AL，[BX]
        CMP     AL，10
        JL      L2
        INC     DX
L2：    INC     BX
        LOOP    L1
```

请回答：（1）**该程序段完成的功能是什么？**

（2）**如果将** JL **改为** JG，**该程序段完成的功能又是什么？**

【解答】（1）**统计** BUF **缓冲区中有符号数大于等于** 10 **的字节个数，其个数放** DX。

（2）**统计** BUF **缓冲区中无符号数小于等于** 10 **的字节个数，其个数放** DX。

17. **设** TAB **中存放的数据为** 30H，31H，32H，33H，34H，35H，36H，37H，38H，39H，**现有程序段如下：**

```
LEA     BX，TAB
MOV     AL，X              ；X 为数字 0~9
XLAT
```

请回答：（1）**该程序段完成的功能是什么？**

（2）**若** X **中的内容为** 4，**则** AL=_____ 。

【解答】（1）**将** X **的数字转换成** ASCII **码。**

（2）AL=34H

18. **现有程序段如下：**

```
        LEA     SI，BUF1
        LEA     DI，BUF2
```

```
        MOV     CX，100
L：     MOV     AL，[SI+100]
        MOV     [DI]，AL
        INC     DI
        DEC     SI
        LOOP    L
```

请回答：该程序段完成的功能是什么？

【解答】将 BUF1 中 100 个字节逆序传送到 BUF2 中。

19. **现有程序段如下：**

```
        MOV     AX，BUF
        CMP     AX，0
        JNS     L
        MOV     BX，BUF+2
        NEG     BX
        MOV     BUF+2，BX
        NOT     AX
        ADC     AX，0
        MOV     BUF，AX
L：     …
```

请回答：该程序段完成的功能是什么？

【解答】将 BUF 缓冲区中的 32 位数据取绝对值存放在 BUF 中。

20. **现有程序段如下：**

```
        MOV     AL，X
        AND     AL，0FH
        JZ      L
        MOV     BYTE  PTR  FLG，1
        JMP     M
L：     MOV     BYTE  PTR  FLG，0
M：     …
```

请回答：该程序段完成的功能是什么？

【解答】检测 X 中的低 4 位是否全为 0，如果是则 1→FLG；否则 0→FLG。

2.4.8　程序设计题

1. **使用串操作指令，将 BUF 缓冲区中的 100 个字节清 0。**

【解答】

```
CLD
        MOV     CX，100
        MOV     AL，0
```

```
        LEA     DI，BUF
        REP     STOSB
```

2. **比较** 5 **个字节的字符串** A **和** B**，若两个字符串不相等则字节标志单元** FLG **置** 1**；否则清** 0**。**

【解答】

```
        CLD
        MOV     CX，5
        LEA     DI，A
        LEA     SI，B
        REPE    CMPSB
        JZ      L
        MOV     BYTE  PTR  FLG，1
        JMP     M
L：     MOV     BYTE  PTR  FLG，0
M：     …
```

3. **统计** BUF **缓冲区** 100 **个字数据中为** 0 **的个数，并将统计的个数存放在** DL**。**

【解答】

```
        MOV     CX，100
        MOV     DL，0
        LEA     BX，BUF
L：     MOV     AL，[BX]
         CMP    AL，0
         JNZ    M
         INC    DL
M：     INC     BX
        LOOP    L
```

4. **计算** Z=（X+5）*Y+30（X，Y **为无符号字节数据）。**

【解答】

```
MOV     AL，X
ADD     AL，5
MUL     Y
ADD     AL，30
ADD     AH，0
MOV     Z，AX
```

5. **将** X：AX：BX **中的** 48 **位数乘以** 2**。**

【解答】

```
SAL     BX，1
RCL     AX，1
RCL     DX，1
```

6. **将有** 100 **个字符的缓冲区** BUF **中的＄符号用空格（**20H**）代替。**

【解答】

```
        MOV     CX，100
        LEA     BX，BUF
```

```
L：  MOV     AL，[BX]
     CMP     AL，‘ $ ’
     JNZ     M
     MOV     AL，20H
     MOV     [BX]，AL
M：  INC     BX
     LOOP    L
```

7. **将有符号的字数据 A 和 B 中的大者存入 C 中。**

【解答】

```
     MOV     AX，A
     CMP     AX，B
     JA      L
     MOV     AX，B
L：  MOV     C，AX
```

8. **检测字单元 A 中的第 4 位是否为 0，是则 FLG 置 1，否则清 0。**

【解答】

```
     MOV     AX，A
     AND     AX，10H
     JZ      L
     MOV     AL，0
     JMP     M
L：  MOV     AL，1
M：  MOV     FLG，AL
```

9. **将 DX：AX 中的 32 位数据逻辑左移 2 位。**

【解答】

```
SHL     AX，1
RCL     DX，1
SHL     AX，1
RCL     DX，1
```

10. **将 100 个元素的字数组 A 中的每个元素减 1。**

【解答】

```
     MOV     CX，100
     LEA     BX，A
L：  INC     WORD PTR [BX]
     ADD     BX，2
     LOOP    L
```

第 3 章　8086 汇编语言程序格式

为了能编好 8086 汇编语言程序，除了要熟练地掌握 8086 的各类指令及指令的各种寻址方式外，还必须熟练地掌握汇编语言的语句格式和源程序的格式等。

在本章的学习中，要求掌握汇编语言的语句格式，汇编语言源程序的格式；理解和掌握各类伪指令的助记符及其作用；操作数的规定、书写格式和作用；了解宏指令的概念、书写格式和作用；熟练地掌握汇编语言程序的上机过程。

【学习重点】

1. 语句的格式。
2. 符号定义语句、数据定义语句和段定义语句。
3. 汇编语言程序的上机过程。

【学习难点】

汇编语言语句的格式和汇编语言源程序的格式。

3.2.1　知识体系结构

在本章中，所讲述的知识体系结构如图 3.1 所示。

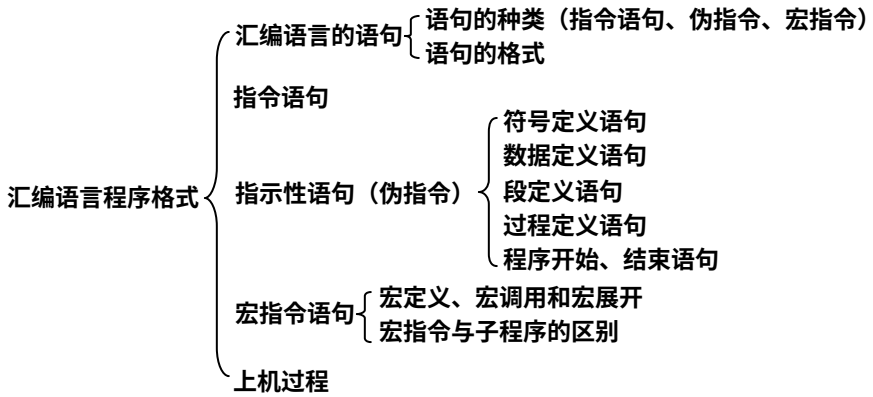


图 3.1　汇编语言程序格式知识体系结构

3.2.2　知识点与考核要求

1. 汇编语言语句的种类和格式，应达到“ 理解 ”的层次。

   （1）汇编语言语句的三种类型（指令语句、伪指令语句和宏指令语句）及在程序中的不同作用。

   （2）组成汇编语言语句的四个部分，每部分内容的规定及相互间的定界符的规定。

2. 指令语句，应达到“ 综合应用 ”的层次。

   （1）指令语句的组成及书写格式。

   （2）指令语句标号的功用和书写格式，指令语句中各种类型操作数的规定。

3. 符号定义语句，应达到“ 简单应用 ”的层次。

   （1）两种符号定义语句的助记符和语句格式；符号名（表达式名）及表达式的内容和格式。

   （2）符号定义语句在编程中的应用。

   （3）标号、变量的三个属性。

   （4）表达式中常用的操作符：

   算术运算符：＋、－、*、/、MOD
   逻辑运算符：AND、OR、XOR、NOT
   关系运算符：EQ、NT、LT、GT、LE、GE
   数值回送符：TYPE、LENGTH、SHORT、THIS、HIGH、LOW

4. 数据定义语句，应达到“ 综合应用 ”的层次。

   （1）三种类型的数据定义语句的助记符和语句格式；变量名定义和使用的规则。

   （2）各种类型操作数的定义，在指令语句和数据定义语句中的使用。

   ① 常用的进位计数制及书写的格式。
   ② ASCII 字符串书写格式。
   ③ 符号名、标号、变量名的使用规则和书写格式。
   ④ 留空单元的书写格式。
   ⑤ 用 DUP 定义的一串数、一串字符或留空单元的书写格式。
   ⑥ 各种常用运算符和操作符的符号、功能、书写格式和使用规则。

5. 段定义语句，应达到“ 简单应用 ”的层次。
   伪指令 SEGMENT 和 ENDS、ASSUME 和 ORG 的功能和书写格式。

6. 过程定义语句，应达到“ 简单应用 ”的层次。

   （1）过程定义语句 PROC 和 ENDP 的功能和书写格式。

   （2）过程定义语句在子程序设计中的应用。

7. 宏指令语句应达到“理解”的层次。

（1）宏指令的作用，了解宏指令语句与过程（子程序）定义语句的异同点。
（2）宏定义、宏调用的书写格式，了解宏汇编后的宏展开。
（3）宏指令中的变元、实元的书写格式和取代规则。

8. 汇编语言源程序格式，应达到“简单应用”的层次。

（1）8086 汇编语言源程序的分段结构的意义、各段的书写格式和书写位置。
（2）能够分析汇编语言源程序，能够按照汇编语言源程序格式编写程序。

9. 汇编语言程序上机过程应达到“识记”的层次。
汇编语言程序上机调试运行的两种方法和步骤。

3.3 例题分析

例 1　用伪指令定义一个字符串变量 MES，其内容为“I　AM　A　STUDENT!”。要求该字符串能在显示器上从新的一行开始显示。

解：因为 MES 是一个字符串变量，可用数据定义伪指令的 DB 对其进行定义。如果要使该字符串从新的一行开始显示，可在该串的前面加回车（0DH）符和换行（0AH）符。如果使用 DOS 功能调用（INT　21H）的 9 号功能——显示该字符串，则应该在字符串的最后以$符号结束。其定义为：

```
MES  DB   0DH，0AH，‘I AM A STUDENT！$’
```

例 2　数据段 DATA 定义如下：

```
DATA   SEGMENT
A      DB      2
B      DW      24H
C      EQU     $－A
D      DB      $＋4
E      DW      B
DATA   ENDS
```

请回答：1. **画出数据段在存储器中的分配示意图。**
2. **写出 A、B、C、D、E 的偏移地址或常数值。**

解：该数据段中的 A 为字节变量，占用一个单元，其值为 2。变量 B 为字变量，其值为 24H，占用两个字节单元，低字节为 24H，高字节为 00H。变量 C 是等价语句定义的，不占内存单元，其值为$-A。由于$符号是汇编指针，它表示汇编到该伪指令后分配内存单元的偏移地址。而该数据段中 A 的偏移地址为 0，汇编指针$此时的值为 3，所以$-A 表示 A、B 两个变量占用的字节数，即 C 为 3。D 变量中的$符号也是汇编指针值，它此时还是指向 3，所以$+4 为 7，所以 D 变量单元中所存的值为 7。而 E 变量中所存储的是

B 变量的偏移地址，即 0001H。由此，该数据段在存储器中的分配示意图如图 3.2 所示。

2
24H
00
7
01
00

图 3.2　例 2 的存储器分配示意图

由于 C 被定义为常量，没有偏移地址，其常量值为 3。A 的偏移地址为 0，B 的偏移地址为 1，D 的偏移地址为 3。E 的偏移地址为 4。

例 3　设数据段定义如下：

```
DATA   SEGMENT
A      DW      23H
B      DB      34H
C      DD      5678H
D      DW      10  DUP (0)
DATA   ENDS
```

请回答：1. 该数据段占用内存多少个字节？

2. 执行“MOV　AL，BYTE　PTR　C”后，AL=____。

解：A 是字，占用 2 个字节单元；B 是字节，占用 1 个字节单元；C 是双字，占用 4 个字节单元；D 是字，共有 10 个字，每字占用 2 个字节单元，D 变量用 20 个字节单元。所以该数据段占用 27 个字节单元。

执行“MOV　AL，BYTE　PTR　C”时，C 是双字变量，而指令是将 C 以字节的形式传送数据，此时指令将 C 的第一个字节内容（78H）传送给 AL，所以 AL=78H。

例 4　判断指令“MOV　BX，OFFSET [SI+3]”是否有错。如果有错则改正。

解：因为 OFFSET 运算符后只能跟标号或变量，不能是表示式，所以该指令是错误的。在取某存储单元的偏移地址时，可用 LEA 指令来实现。所以上述指令可改为：

```
LEA  BX，[SI+3]。
```

例 5　根据下列要求设置数据段 DATA。

BUF 为 10 字节的缓冲区，内容不定。
BCDB 为十进制数字节变量：45。
TAB 为 ASCII 码的字节变量：5678。
MES 为字符串变量：‘personal computer’
C 为常量：100
D 为字变量：TAB 的偏移地址。

解：BUF **是字节常量，由于内容不定，可用重复子句** n DUP（？）**表示；**BCDB **为字节变量，由于只占用一个字节的数据，所以可用** BCDB DB 45H **表示；**TAB **为** ASCII **码的字节变量，而** 5678 **是** 4 **位十进制数，需占用** 4 **个存储单元来存放，即** 35H、36H、37H、38H**；**MES **是一个字符串变量，需用** DB **定义‘** personal computer **’；**C **是常量，可用等价语句或等号语句定义即** C=100（**或** C EQU 100）**；**D **为字变量，用** DW **定义，其值为** TAB **的偏移地址，可用** D DW TAB **表示。所以数据段定义如下：**

```
DATA    SEGMENT
BUF     DB      10  DUP （?）
BCDB    DB      45H
TAB     DB      35H，36H，37H，38H
MES     DB      ‘personal computer’
C       =       100
D       DW      TAB
DATA    ENDS
```

例 6　**现有程序如下：**

```
DATA   SEGMENT
BUF    DB      100   DUP（?）
SUN    DW      0
DATA   ENDS
CODE   SEGMENT
       ASSUME  CS：CODE，DS：DATA
START：MOV     AX，DATA
       MOV     DS，AX
       LEA     BX，BUF
       MOV     CX，100
       MOV     AX，0
L：    ADD     AL，[BX]
       ADC     AH，0
       INC     BX
       LOOP    L
       MOV     SUN，AX
       MOV     AH，4CH
       INT     21H
CODE   ENDS
       END     START
```

请回答：（1）**该程序段完成什么功能？**

（2）**为什么在程序中要用“** ADC　AH，0 **”指令？**

解：该程序中有两个变量 BUF **和** SUN**。其中** BUF **有** 100 **个字节的数据，该程序是将** BUF **中的数据依次加到** AX **寄存器中，共计循环相加** 100 **次。即完成了** BUF **缓冲区中** 100 **个字节的无符号数相加，结果存放在** SUN **中。**

如果在程序中不用“ADC　AH，0”指令，当有多个字节相加时，可能会产生溢出。而使用该指令后，将每次溢出的位加到 AH 中，使相加后的结果由原来的 8 位数变成 16 位数，保证了结果不会出现溢出错误。

3.4 练习题与参考答案

3.4.1 单项选择题

1. 下列选项中不能作为名字的是（　　）。

A. FH　　B. A3　　C. 3B　　D. FADC

2. 下列指令不正确的是（　　）。

A. MOV　AL，123　　B. MOV　AL，123Q
C. MOV　AL，123D　　D. MOV　AL，123H

3. 下列指令不正确的是（　　）。

A. MOV　BL，OFFSET　A　　B. LEA　BX，A
C. MOV　BX，OFFSET　A　　D. MOV　BX，A

4. 若定义“BUF　DB　1，2，3，4”，执行“MOV　AL，TYPE　BUF”后 AL=（　　）。

A. 0　　B. 1　　C. 2　　D. 3

5. 若定义“A　EQU　100”，执行“MOV　AX，A”后，AX=（　　）。

A. A 的偏移地址　　B. A 单元中的内容
C. 100　　D. A 的段地址

6. 若定义“B　DW　1，2，10　DUP（0）”，则该伪指令分配（　　）个字节单元。

A. 10　　B. 20　　C. 22　　D. 24

7. 若定义“C　DD　2，4”，则该伪指令分配（　　）个字节单元。

A. 2　　B. 4　　C. 6　　D. 8

8. 伪指令是（　　）规定的汇编说明助记符，它在源程序进行汇编时说明。

A. DEBUG　　B. LINK　　C. MASM　　D. EDIT

9. 在上机操作过程中，执行 MASM 命令后，除了生成一个目标文件外，根据选择还可以生成一个（　　）文件。

A. .LST　　B. .EXE　　C. .MAP　　D. .ASM

10. 执行 LINK 命令后可以生成一个以（　　）为扩展名的文件。

A. ASM　　B. EXE　　C. OBJ　　D. COM

11. 一个段最大可定义（　　）个字节。

A. 1M　　B. 64K　　C. 32K　　D. 16K

12. 若要求一个段的起始地址从能被 256 整除的单元开始，在定位方式选项中应选（　　）。

A. BYTE　　B. WORD　　C. PARA　　D. PAGE

13. 宏指令与子程序相比，在多次调用时，宏指令调用的目标程序长度比子程序调用的目标程序（　　）。

A. 相同　　B. 长　　C. 短　　D. 不定

14. 宏指令与子程序相比，子程序调用程序的执行速度比宏指令（　　）。

A. 相同　　B. 快　　C. 慢　　D. 不定

15. ASSUME 伪指令说明了汇编程序所定义的段与段寄存器的关系，它只影响（　　）的设定。

A. 源程序　　B. 目标程序　　C. 汇编程序　　D. 连接程序

16. 设 A 和 B 为字变量，C 为标号，下列指令中不正确的是（　　）。

A. MOV　AX，A　　B. MOV　AX，B
C. JNE　A　　D. JMP　C

17. 代码段中的语句（　　）表示该段结束。

A. ASSUME　　B. CODE　ENDS
C. START：MOV　AX，DATA　　D. END　START

18. 过程定义语句以“过程名 PROC”开始，以过程名（　　）结束。

A. ENDS　　B. ENDP　　C. ENDM　　D. END

19. 一个段可以放在内存的任何地方，但起始地址应该从一个能被（　　）整除的单元地址开始。

A. 16　　B. 32　　C. 64　　D. 128

20. 在数据定义语句中，下列描述不正确的是（　　）。

A. 存放存储单元的地址可以用字节表示
B. 存放存储单元的地址可以用字表示
C. 存放存储单元的地址可以用双字表示

D. 存放存储单元的地址可以用四字表示

单项选择题参考答案

1. C	2. D	3. A	4. B	5. C	6. D
7. D	8. C	9. A	10. B	11. B	12. D
13. B	14. C	15. A	16. C	17. B	18. B
19. A	20. A				

3.4.2　多项选择题

1. 在 8086 汇编语言中，语句的种类有（　　）。

A. 指令性语句　　B. 指示性语句
C. 汇编语句　　D. 说明语句

2. 在下列的选项中，不能作为名字的有（　　）。

A. AX　　B. 3MA　　C. ABCD　　D. MOV

3. 在下列的选项中，作为变量的类型有（　　）。

A. 字节　　B. 字　　C. 双字　　D. 近程

4. 在下列的选项中，作为标号的类型有（　　）。

A. DB　　B. DD　　C. NEAR　　D. FAR

5. 在运算符 OFFSET 后可以是（　　）。

A. 标号　　B. 变量　　C. 表达式　　D. 数字

6. 在指令“LEA　BX，X”中的 X 可以是（　　）。

A. 标号　　B. 变量　　C. 表达式　　D. 数字

7. 不能分配内存单元的伪指令语句有（　　）。

A. EQU　　B. DW　　C. =　　D. DD

8. 定义一个段的伪指令语句有（　　）。

A. NAME　SEGMENT　　B. NAME　PROC
C. NAME　ENDP　　D. NAME　ENDS

9. 定义一个过程的伪指令语句有（　　）。

A. NAME　SEGMENT　　B. NAME　PROC
C. NAME　ENDP　　D. NAME　ENDS

10. 宏指令与子程序的主要区别在于（　　）。

A. 完成的功能完全不同　　B. 目标程序的长度不同
C. 执行程序的速度不同　　D. 汇编时处理的方式不同

多项选择题参考答案

1. AB	2. ABD	3. ABC	4. CD	5. AB
6. ABC	7. AC	8. AD	9. BC	10. BCD

3.4.3 填空题

1. 汇编语言的语句有指令语句和_____，宏指令是_____的另一种形式。
答：指示性语句（伪指令）　指令语句

2. 标号的三个属性是段地址、_____和类型。
答：偏移地址

3. 变量的三个属性是段地址、_____和类型。
答：偏移地址

4. 标号的类型有_____和_____　。
答：近程（NEAR）　远程（FAR）

5. 变量的类型有_____、_____、_____、四字（八字节）和十字节。
答：字节（BYTE）　字（WORD）　双字（DWORD）

6. 等价语句不能重复定义、等号语句_____重复定义。
答：可以

7. 等价语句和等号语句_____内存单元。
答：不分配

8. 语句“MOV　BX，OFFSET　A”可用_____一条指令来代替。
答：LEA　BX，A

9. 若定义“A　DW　1234H”，指令“MOV　AL，BYTE　PTR　A”执行后，AL=_____。
答：34H

10. 若定义“B　DW　1，2，30 DUP（5）”，则 B 分配的内存单元数是_____个字节。
答：64

11. 若定义“C　DW‘AB’”，则指令“MOV　AL，BYTE　PTR　C”执行后，AL=_____。
答：42H

12. 在程序的操作数项中使用的段名、标号名、变量名和符号名都必须在源程序中先_____，否则汇编程序进行汇编时就会_____。

答：定义　报错

13. 使用伪指令_____和_____定义一个段。

答：“段名　SEGMENT”　“段名　ENDS”

14. 使用伪指令_____和_____定义一个过程。利用过程定义语句可以把程序分成小段，以便于_____、_____、调试和修改。

答：“过程名　PROC”　“过程名　ENDP”，阅读，理解

15. 伪指令_____表示整个程序结束。

答：“END　[表达式]”

16. 宏展开就是用宏定义取代源程序中的宏指令。若实元个数大于变元个数，则多余的实元_____。若实元个数少于变元个数，则多余的变元用_____代替。

答：不予考虑　空格

17. 汇编语言源程序经_____后产生目标文件，它_____直接在机器上运行，还必须经过_____后形成可执行文件。

答：MASM（汇编）　不能　LINK（连接）

18. 在数据段中使用字符串时，该字符必须用_____括起来。当定义含有多个字符的字符串时，只能使用_____伪指令。

答：引号　DB

19. 用汇编语言编写的程序称为_____，其扩展名为_____。

答：汇编语言源程序　ASM

20. 在源程序中只要对某一具有独立功能的程序段进行一次宏定义，就可以_____次调用它。

答：多

3.4.4　简答题

1. 按照下列题目要求写出每小题的伪指令。

（1）将 12H、34H、56H、78H 存放在字节变量 A 的存储单元中。

（2）将 12、1638H、0E52H 存放在字变量 B 的存储单元中。

（3）将字符串 COMPUTER 存放在 C 变量的存储单元中。

（4）将 D 字节变量的 100 个单元设置为 0。

（5）将 D 的偏移地址存放在 E 变量中。

【解答】（1）A　DB　　12H，34H，56H，78H

（2）B　DW　　12，1638H，0E52H

（3）C　DB　　‘COMPUTER’

（4）D　DB　　100　DUP（0）

（5）E　DW　　D

2. **设数据段定义如下：**

```
DATA    SEGMENT
BUF1    DB      2，3
BUF2    DW      4，5，6
BUF3    DB      2，100  DUP（0）
BUF4    DB      ‘ABCDE’
BUF5    DW      BUF3
BUF6    EQU     $-BUF3
DATA    ENDS
```

请回答：（1）**该数据段占用的内存有多少个字节？**

（2）BUF5 **单元中的内容是多少？**

（3）BUF6 **的内容是多少？**

（4）**执行“**MOV　AL，BUF4+2**”指令后，**AL=_____。

（5）**执行“**MOV　AX，WORD　PTR　BUF1**”指令后，**AX=____。

【解答】（1）116

（2）0008H

（3）108

（4）43H

（5）0302H

3. **设数据段定义如下：**

```
DATA    SEGMENT
A       DW      23，45
B       DW      ‘EF’
C       =       $+5
D       DB      10  DUP（?）
E       EQU     $－D
F       DW      D
DATA    ENDS
```

请回答：（1）**该数据段占用的内存有多少个字节？**

（2）C **的值为多少？**

（3）E **的值为多少？**

（4）**执行“**MOV　AX，F**”指令后，**AX=_____。

（5）**执行“**MOV　AL，BYTE　PTR　B**”指令后，**AL=_____。

【解答】（1）18

（2）11

（3）10

（4）0006H

（5）46H

4. **设数据段定义如下：**

```
DATA    SEGMENT
BUF1    DB      36H，3，2
BUF2    DW      1364H，253H
BUF3    DB      30  DUP（'ABC'）
BUF4    DW      $-BUF3
BUF5    DB      100  DUP（0）
DATA    ENDS
```

请回答：（1）**执行“**MOV　AX，WORD　PTR　BUF1**”后，**AX=_____。

（2）**执行“**LEA　BX，BUF3**”后，**BX=_____。

（3）**执行“**MOV　CX，BUF4**”后，**CX=_____。

（4）**执行“**MOV　AL，BUF3+2**”后，**AL=_____。

（5）**执行“**MOV　AX，BUF2+1**”后，**AH=_____。

【解答】（1）0336H

（2）0007H

（3）90

（4）43H

（5）13H

5. **设数据段定义如下：**

```
DATA    SEGMENT
A1      DB      10  DUP（  ）
A2      DW      1234H，5678H，341H
A3      DW      $-A2
A4      DW      A2
A5      DB      7，3，2
DATA    ENDS
```

请回答：（1）**用一条指令将** A2 **的偏移地址送** BX**。**

（2）**将字符‘**A**’的** ASCII **码送** A1 **的第** 6 **个字节**

（3）**将** A2 **的第** 3 **个字节的内容送** AL**。**

（4）**将** A2 **的第** 3 **个单元开始的字内容送** AX**。**

（5）**用一条伪指令求出** A1 **和** A2 **所占字节数（设伪指令的变量为** C**）。**

【解答】（1）LEA BX，A2　**或**　MOV　BX，OFFSET A2

（2）MOV　A1+5，BYTE　PTR　41H

（3）MOV　AL，BYTE　PTR　A2+2

（4）MOV　AX，A2+2

（5）C　EQU　A3-A1　**或**　C　=　A3-A1

6. **设数据段定义如下：**

```
DATA    SEGMENT
A       DW      2，3
B       DB      100  DUP（0）
C       DD      12345678H
D       DB      $-B
E       DW      B
DATA    ENDS
```

请回答：（1）**用一条指令将** C **的第** 2 **个字节的内容取出并送** AL。

（2）**执行完“** MOV　AL，LENGTH　B**”后，**AL= _____。

（3）**执行完“** MOV　AL，TYPE　C**”后，**AL= _____。

（4）**执行完“** MOV　AL，LENGTH　A**”后，**AL= _____。

（5）**执行完“** MOV　AL，SIZE　B**”后，**AL= _____。

（6）**执行完“** MOV　AL，D**”后，**AL= _____。

（7）**执行完“** MOV　BX，E**”后，**BX= _____。

（8）**执行完“** MOV　AL，A＋2**”后，**AL= _____。

【解答】（1）MOV　AL，BYTE　PTR　C+1

（2）100

（3）4

（4）1

（5）1

（6）104

（7）0004

（8）03H

7. **设数据段定义如下：**

```
DATA    SEGMENT
A       DW      1，2，3
B       DB      ‘ABCDEF’
C       DW      ‘AB’
D       DB      12H，34H，56H
E       =       $+5
DATA    ENDS
```

请回答：（1）**用一条指令将** B **字符串中的字符‘** E **’送** AL。

（2）**用一条指令将** D **变量中的第** 2 **个、第** 3 **个字节送** AX。

（3）**用一条指令将** A **变量中的第** 3 **个字节送** AL。

（4）**“** MOV　AX，C**”执行后，**AX=_____。

（5）“MOV　AL，E”执行后，AL=_____。

【解答】（1）MOV　AL，B+4
（2）MOV　AX，WORD　PTR　D+1
（3）MOV　AL，BYTE　PTR　A+2
（4）4142H
（5）16H

8. 按下列要求写出数据段 DATA 的相应内容。

（1）TAB 是 0～9 的 ASCII 码表。
（2）STR 是字符串变量，字符串为 COMPUTER。
（3）BUF 是 100 个空格的缓冲区。
（4）COUNT 是计算 BUF 缓冲区中字节数的常量变量。
（5）AD 是存放 TAB 的偏移地址变量。
（6）D 是字节变量，数值为 15H，2，3

【解答】
```
DATA    SEGMENT
TAB     DB      '0123456789'
STR     DB      'COMPUTER'
BUF     DB      100  DUP ('  ')
COUNT   =       $-BUF
AD      DW      TAB
D       DW      15H，2，3
DATA    ENDS
```

9. 按下列要求写出数据段 DSEG 的相应内容。

（1）STR 是存放字符 ABCDE 的字符串变量。
（2）D 是存放十进制 BCD 数据 372，2673，6852 的字变量。
（3）BUF 是存放 10 个 0 的字节缓冲区。
（4）E 是存放 BUF 偏移地址和段地址的双字变量。
（5）F 是常量，其值为 5。

【解答】
```
DSEG    SEGMENT
STR     DB      'ABCDE'
D       DW      372H，2673H，6852H
BUF     DB      10  DUP (0)
E       DD      BUF
F       =       5
DSEG    ENDS
```

10. 简述宏指令与子程序的主要区别。

【解答】汇编程序处理宏指令时，是把宏定义的宏体插入到宏调用处，有多少次调用就插入多少次。所以宏指令并没有简化目标程序，它相对于程序调用而言，比子程序调用

占用的内存单元多。

汇编程序处理子程序时，每次调用并不需要把子程序的代码插入到主程序中，大大地节省了内存空间，但每次调用子程序时都要转子程序，用来保护现场，调用结束后还要恢复现场并返回，显然，它花费的时间比宏调用花费的时间多。

如果替代的程序段不长，调用的次数不多，以速度为主要考虑因素时，通常采用宏指令。如果替代的程序段较长，以节省内存空间为主要考虑因素时，通常采用子程序。

11. 在操作系统状态下，执行 MASM 程序对某一汇编语言源程序进行汇编后，除产生目标文件外，通过对答方式还可以产生另外对应源程序的两个文件。请指出：

（1）这两个文件的扩展名是什么？

（2）这两个文件的功能是什么？

【解答】（1）列表文件（扩展名为 .LST）和交叉引用文件（扩展名为 .CRF）

（2）列表文件给出了源程序中的每条语句及其对应的目标代码，以及每条语句在段内的偏移地址，以便使用者检查和阅读。交叉引用文件给出了源程序中定义的符号值和程序中引用这些符号的基本情况。

12. 简述汇编程序对宏调用作宏展开的过程。

【解答】宏展开就是用宏定义的宏体取代源程序中的宏指令，同时用宏调用中的实元来取代定义中的变元。在取代时，实元与变元一一对应时，就将第一个实元取代第一个变元，第二个实元取代第二个变元，其他依此类推。当实元个数多于变元个数时，则多余的实元不予考虑，当实元的个数少于变元个数时，则多余的变元作空处理。

13. 设数据段定义如下：

```
DATA    SEGMENT
A       DW      1，2
B       DB      3，4
C       DB      ‘ABCD’
D       =       $-C
E       DB      3 DUP（0）
DATA    ENDS
```

请回答：（1）画出该数据段在内存中的分配示意图。

（2）变量 A、B、C、E 的偏移地址各是多少？

（3）变量 D 的值是多少？

【解答】（1）示意图如图 3.3 所示。

（2）A 的偏移地址为 00H；

B 的偏移地址为 04H；

C 的偏移地址为 06H；

E 的偏移地址为 0AH。

（3）D 的值为 4。

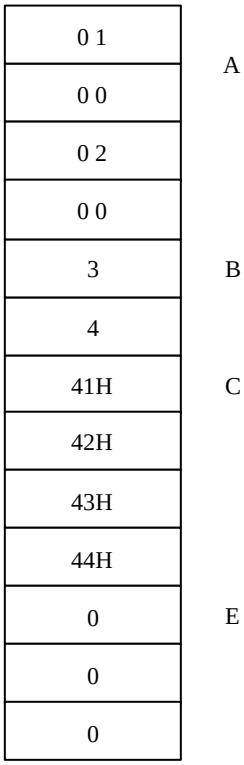


图 3.3　数据段在内存中的分配

14. **设数据段定义如下：**

```
DATA    SEGMENT
BUF1    DW      123H
BUF2    DB      'SIMPLE'
BUF3    DW      $+4
BUF4    EQU     $-BUF1
DATA    ENDS
```

请回答：（1）**画出该数据段在内存中的分配示意图。**

（2）BUF1、BUF2、BUF3 **的偏移地址各是多少？**

（3）BUF4 **的值是多少？**

（4）**“** MOV　AL，BUF2+5 **”执行后，**AL=_____**。**

【解答】（1）**示意图如图** 3.4 **所示。**

（2）BUF1 **的偏移地址为** 00H**；**

BUF2 **的偏移地址为** 02H**；**

BUF3 **的偏移地址为** 08H**。**

（3）BUF4 **的值为** 10**。**

（4）45H**。**

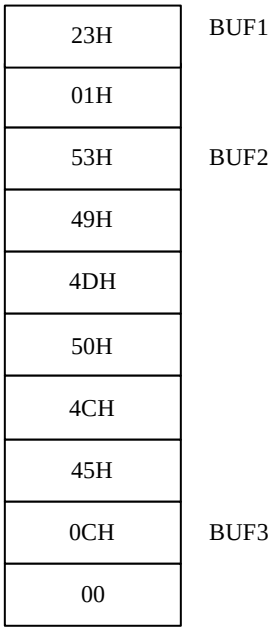


图 3.4　**数据段在内存中的分配**

15. **设数据段定义如下：**

```
DATA    SEGMENT
        ORG=4
A       EQU     100
B       DB      ‘ABC’，2
C       DW      101B
D       EQU     B+4
        DATA    ENDS
```

请回答：（1）**画出该数据段在内存中的分配示意图。**

（2）**B、C 的偏移地址各是多少？**

（3）**“** MOV　AX，D **”执行后，**AX=_____**。**

【解答】（1）**示意图如图** 3.5 **所示。**

（2）**B 的偏移地址为** 04H**；**

C **的偏移地址为** 08H**。**

（3）0008H

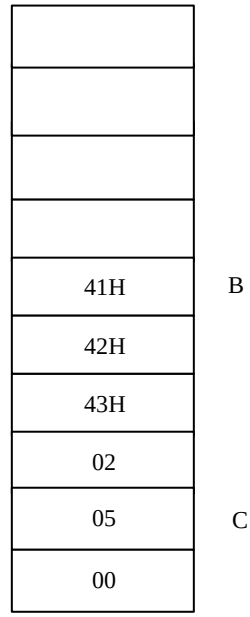


图 3.5　数据段在内存中的分配

3.4.5　判断改错题

1. **设数据段定义如下，判断各语句是否正确，如有错误则改正。**

```
DATA     SEGMENT
A        DB      123H
B        DW      23，45678H
C        DW      ‘ABCD’
D        DB      100  DUP (‘ABC’)
E        DB      ($-D) /3
         DATA    END
```

答：(1) **段定义语句的开始正确，而结束语句** DATA　END **错，应改为** DATA　ENDS。

(2) A **变量的定义错，因为** 123H **超出一个字节，应改为** DW。

(3) B **变量的定义错，因为** 45678H **超出一个字，应改为** DD。

(4) C **变量的定义错，因为多字符的字符串应用** DB。

(5) D **变量的定义正确。**

(6) E **变量的定义正确。**

2. **设数据段定义如下：**

```
DATA     SEGMENT
BUF1     DB      2，3
BUF2     DW      3
BUF3     DD      5
BUF4     EQU     $-BUF1
```

```
BUF5    DB      'ABCD'
BUF6    DW      BUF5
DATA    ENDS
```

请判断下列指令是否正确，如果有错请改正。

（1）MOV　AX，BUF1
（2）MOV　AL，BUF5
（3）MOV　AX，BUF4
（4）LEA　BL，BUF2
（5）MOV　AX，BUF3
（6）MOV　[BX]，TYPE　BUF3
（7）MOV　[BX]，WORD　PTR　BUF3
（8）MOV　BUF5，AL
（9）MOV　BUF6，BUF2
（10）MOV　BX，BUF6

答：（1）**错，**MOV　AX，WORD　PTR　BUF1
（2）**对**
（3）**对**
（4）**错，**LEA　BX，BUF2
（5）**错，**MOV　AX，WORD　PTR　BUF3
（6）**错，**MOV　BYTE　PTR[BX]，TYPE　BUF3
（7）**对**
（8）**对**
（9）**错，**MOV　AX，BUF2
　　MOV　BUF6，AX
（10）**对**

3. **判断下列伪指令是否正确，如果有错请改正。**

（1）DATA　SEG
　　M
　　ENDS
（2）CODE　SEGMENT
　　M
　CODE　END
（3）MA　SEGMENT
　　M
　　ENDM
（4）STACK　SEGMENT　'STACK'

（5）
```
A       SEGMENT
B       DW    1
C       DB     123H
A       ENDS
```
（6）
```
MYNAME    SEGMENT    PARA
                       M
MYNAME    ENDS
```

答：（1）**错，**
```
DATA    SEGMENT
                M
        DATA    ENDS
```
（2）**错，**
```
CODE    SEGMENT
                M
        CODE    ENDS
```
（3）**错，**
```
MA    PROC
          M
        ENDM
```
（4）**错，**STACK　SEGMENT　STACK　‘STACK’。

（5）**错，将** C　DB　123H　**改为**　C　DW　123H。

（6）**对。**

3.4.6　程序分析题

1. **分析下列程序，说明程序的功能。**

```
DATA     SEGMENT
A        DB      18H, 34H, 05H, 06H, 09H
         DB      0AH, 0CH, 11H, 12H, 14H
B        DB      3, 4
C        DB      2 DUP (0)
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS:CODE, DS:DATA
START:   MOV     AX, DATA
         MOV     DS, AX
         LEA     BX, A
         MOV     CX, 2
         MOV     SI, OFFSET  B
         LEA     DI, C
L:       MOV     AL, [SI]
         XLAT
         MOV     [DI], AL
         INC     SI
```

```
        INC     DI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】从表 A 中查找 B 变量中的两个数的对应的值送 C 变量中保存。

2. **分析下列程序，说明程序的功能。**

```
DATA    SEGMENT
TAB     DB      30H，31H，32H，33H，34H，35H
        DB      36H，37H，38H，39H
STR     DB      3，2，7，6，8
COUNT   EQU     $-STR
BUF     DB      10  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     BX，TAB
        LEA     DI，BUF
        MOV     CX，COUNT
        LEA     SI，STR
L：      MOV     AL，[SI]
        XLAT
        MOV     [DI]，AL
        INC     SI
        INC     DI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】将 STR 中的十进制数经过查表转换成 ASCII 码并送 BUF 保存。

3. **分析下列程序，说明程序的功能。**

```
DATA    SEGMENT
A       DB      12，34，56H，0
B       DB      53H，62，31H，0
C       DB      4 DUP (0)
DATA    ENDS
CODE    SEGMENT
```

```
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，A
        LEA     DI，B
        LEA     BX，C
        MOV     CX，4
        CLC
L：      MOV     AL，[SI]
        ADC     AL，[DI]
        MOV     [BX]，AL
        INC     SI
        INC     DI
        INC     BX
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】将 A 中的 4 个字节数与 B 中的 4 个字节数（多精度）相加，并将结果放在 C 中。

4. **分析下列程序，说明程序的功能。**

```
DATA    SEGMENT
A       DW      ?
B       DW      ?
C       DW      ?
D       DW      ?
E       DW      2 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，A
        MOV     DX，0
        ADD     AX，B
        ADC     DX，0
        ADD     AX，C
        ADC     DX，0
        MOV     CX，D
        DIV     CX
        MOV     E，AX
        MOV     E+2，DX
```

```
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】计算 16 **位无符号数（**A+B+C）/D **的表达式，结果的商放在** E **中，余数放在** E+2 **中。**

5. **分析下列程序，说明程序的功能。**

```
DATA    SEGMENT
A       DB      36H
B       DB      0DH，0AH，‘NUM=’
C       DB      4  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        LEA     BX，C
        MOV     CL，4
        SHR     AL，CL
        AND     AL，0FH
        CMP     AL，9
        JNA     N
        ADD     AL，7
N：      ADD     AL，30H
        MOV     [BX]，AL
        INC     BX
        MOV     AL，A
        AND     AL，0FH
        CMP     AL，9
        JNA     M
        ADD     AL，7
M：      ADD     AL，30H
        MOV     [BX]，AL
        INC     BX
        MOV     BYTE  PTR[BX]，‘H’
        INC     BX
        MOV     BYTE  PTR[BX]，‘$’
        LEA     DX，B
        MOV     AH，9
        INT     21H
        MOV     AH，4CH
        INT     21H
```

```
CODE    ENDS
        END     START
```

【解答】将 A 字节变量的数转换成 2 位十六进制的 ASCII 码送 C 中保存，并在显示器上显示出 NUM=36H。

6. **分析下列程序。**

```
DATA    SGEMENT
A       DW      123H，456H，789H
B       DB      12H，34H，56H，78H
C       DB      10 DUP（2，3）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，A
        LEA     DI，C
        MOV     CX，10
L：      MOV     AL，[SI]
        MOV     [DI]，AL
        INC     SI
        INC     DI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：C 的前 5 个字节单元中的内容依次是什么？

【解答】C 前面的 5 个字节单元中的内容依次为 01H，23H，04H，56H，07H

7. **分析下列程序。**

```
DATA    SEGMENT
A       DB      83H，62H，56H，0FAH，67H
B       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     BX，A
        MOV     CX，5
L：      MOV     AL，[BX]
```

```
        CMP     AL，100
        JAE     M
        INC     BYTE  PTR  B
M：     INC     BX
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序运行后，B 中的内容是多少？

【解答】（1）统计 A 缓冲区中小于 100 的无符号数，B 中存放统计的个数。

（2）2。

8. **分析下列程序。**

```
DATA    SEGMENT
A       DB      23H
B       DB      67H
C       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        CMP     AL，B
        JGE     L
        XCHG    AL，B
        MOV     A，AL
L：     MOV     C，AL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序执行后，A、B、C 变量中的值各为多少？

【解答】（1）比较 A 和 B 中的两个有符号数，将较大数存放于 A 和 C 中，较小数存放于 B 中。

（2）（A）=67H，（B）=23H，（C）=67H

9. **宏定义语句如下：**

```
M     MACRO  X, Y, Z
      MOV DX, X
      MOV AH, Y
      INT     Z
ENDM
```

宏调用语句如下：

```
M     BUF1, 9, 21H
M     BUF2, 9, 21H
```

请写出上述两条宏调用的宏展开语句。

【解答】

```
+    MOV     DX, BUF1
+    MOV     AH, 9
+    INT     21H
+    MOV     DX, BUF2
+    MOV     AH, 9
+    INT     21H
```

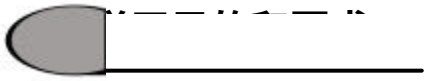
10. **宏定义语句如下：**

```
W    MACRO   X, Y, Z
     LOCAL   L
     LEA     BX, X
     MOV     CX, Y
     MOV     AX, 0
L:   ADD     AL, [BX]
     ADC     AH, 0
     INC     BX
     LOOP    L
     MOV     Z, AX
     ENDM
```

请回答：执行“W　BUF，100，C”指令后，宏调用的程序功能是什么？

【解答】将 BUF 中的 100 个字节的内容相加，并将相加的结果存放在 C 中。

第 4 章　顺序程序设计



顺序程序设计是汇编语言程序设计的方法之一。顺序程序是程序结构形式中最简单、最常用的程序结构形式。它是组成其他复杂程序的基础。学习顺序程序设计的编程方法，是为编写复杂程序打下良好的基础。

通过本章的学习，要求深刻理解程序设计的基本概念；充分认识程序设计方法的重要性和必要性；熟练地掌握顺序程序的结构形式、顺序程序设计的基本步骤和基本方法。

【学习重点】

顺序程序的基本结构和顺序程序设计的基本方法。

【学习难点】

程序的流程图和算法。

4.2.1　知识体系结构

在本章中，所讲述的知识体系结构如图 4.1 所示。

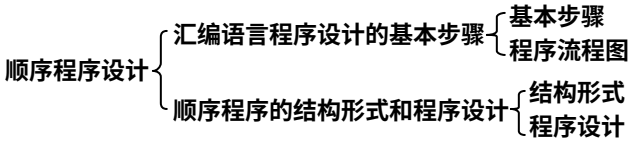


图 4.1　顺序程序设计知识体系结构

4.2.2　知识点与考核要求

1. 汇编语言程序设计的基本步骤，应达到“理解”的层次。

（1）汇编语言程序设计中基本设计步骤的必要性和重要性。

（2）程序设计基本步骤中的每一步骤的要点及达到的目标。

2. **算法和程序流程图，应达到“简单应用”的层次。**

（1）算法和程序流程图的概念及它们在程序设计过程中的重要地位。

（2）根据实际问题列出算法。

（3）根据算法画出程序流程图，理解流程图的基本符号的含义和画法。

3. **顺序程序的基本结构，应达到“简单应用”的层次。**

什么是顺序程序，顺序程序的结构形式。

4. **顺序程序的设计，应达到“综合应用”的层次。**

（1）分析问题、确定算法、画出程序流程图、分配工作单元和选择合适指令编写程序的设计思想和方法。

（2）能够分析顺序程序，能够用顺序程序设计方法编写程序。

4.3 例题分析

例 1　设 X、Y 为无符号数，X 为字变量，Y 为字节变量，下列程序段的功能是完成计算表达式 S=X+（Y+5）×4，请按照题目要求在程序的空格处填写适当的指令。

```
DATA      SEGMENT
X         DW      ?
Y         DB      ?
S         DW      2 DUP (0)
DATA      ENDS
CODE      SEGMENT
          ASSUME  CS:CODE, DS:DATA
START:    MOV     AX, DATA
          MOV     DS, AX
          MOV     AH, 0
          MOV     AL, Y
          ____(1)____
          ____(2)____
          MOV     BX, 4
          ____(3)____
          ADD     AX, X
          ____(4)____
          MOV     S, AX
          ____(5)____
          MOV     AH, 4CH
          INT     21H
CODE      ENDS
          END START
```

解：要计算表达式 S=X+（Y+5）×4 **的值，一般的算法是根据表达式运算符的优先级顺序进行计算。该表达式可以分解成四个步骤来进行计算。第一步是计算** Y+5，**第二步是计算**（Y+5）×4，**第三步是计算** X+（Y+5）×4，**第四步是存放运算结果。在第一步的运算中，由于** Y **是字节，计算** Y+5 **时可能会产生溢出。为了解决这一问题，应将字节相加之后，将可能产生的进位加到高字节** AH **寄存器中，所以在**（1）、（2）**处空格应填“** ADD AL，5 **”和“** ADC　AH，0 **”两条指令。在第二步的计算中，由于第一步** Y+5 **的计算结果在** AX **中，要计算**（Y+5）×4，**应先将** 4 **送往某** 16 **位的寄存器，然后再做乘法运算。根据题意，该运算的数据都是无符号数，所以第**（3）**空应填“** MUL　BX **”，所得结果存放在** DX：AX **中。在第三步的运算中，要将** X **的** 16 **位数与** DX：AX **中的** 32 **位数相加，则必须先进行低** 16 **位数的相加，然后再进行高** 16 **位数的带进位加。由此可知，第**（4）**空应填“** ADC　DX，0 **”。第四步，因为已将运算结果的低** 16 **位送** S **字单元，应将高** 16 **位送** S+2 **字单元，所以第**（5）**空填“** MOV S+2，DX **”。**

例 2　**下列程序的功能是将** BUF **缓冲区中的一个字节的压缩型** BCD **码转换成** 2 **位十进制数的** ASCII **码，并将转换的结果分别按高位和低位送显示器显示。请在程序的空格处填写相应的指令。**

```
DATA      SEGMENT
BUF       DB      26H
DATA      SEGMENT
          ASSUME  CS：CODE，DS：DATA
START：   MOV     AX，DATA
          MOV     DS，AX
          MOV     DL，BUF
          MOV     CL，4
          ____(1)____
          ____(2)____
          MOV     AH，2
          INT     21H
          MOV     DL，BUF
          ____(3)____
          ADD     DL，30H
          MOV     AH，2
          INT     21H
          MOV     AH，4CH
          INT     21H
CODE      ENDS
          END     START
```

解：要完成该程序的功能，必须将 BUF **缓冲区中的** 8 **位二进制数分解成高** 4 **位数和低** 4 **位数分别进行转换。在转换的过程中，第一步是先将** BUF **缓冲区中的高** 4 **位数右移到低** 4 **位，然后根据低** 4 **位中的值转换成对应的** ASCII **码，所以第**（1）**空应填“** SHR　DL，CL **”。由于** 4 **位二进制数表示的** BCD **码是** 0~9，**而** 0~9 **的** ASCII **码是** 30H~39H，**只需在原**

有的数字上加 30H 就可得到对应的 ASCII 码，所以第（2）空应填“ ADD　DL，30H”。同理，在实现低 4 位数的转换时，必须先将高 4 位的值清 0，然后才能进行字符的转换。所以在第（3）空应填“ AND　DL，0FH”。

例 3　现有程序如下：

```
DATA      SEGMENT
 X        DB      4，3
 Y        DB      8，5
 Z        DB      5，6
 R        DW      0
 C        DB      2
 DATA     ENDS
 CODE     SEGMENT
          ASSUME  CS：CODE，DS：DATA
START：   MOV     AX，DATA
          MOV     DS，AX
          MOV     AH，0
          MOV     AL，X
          ADD     AL，Y
          AAA
          MOV     DL，AL
          MOV     AL，X+1
          ADC     AL，Y+1
          AAA
          XCHG    DL，AL
          SUB     AL，Z
          AAS
          XCHG    AL，DL
          SBB     AL，Z+1
          AAS
          MOV     DH，AL
          MOV     AX，DX
          AAD
          CBW
          IDIV    C
          MOV     R，AX
          MOV     AH，4CH
          INT     21H
CONE      ENDS
          END     START
```

请回答：（1）该程序实现什么表达式的计算？

（2）程序运行后 R 中的内容为_____。

解：该程序中，首先将 X 中的 4 和 Y 中的 8 相加，经过 AAA 的调整后，AL 中的内容为 2，而进位位 CF=1，表示两数相加后个位数有向高位（十位数）的进位。然后将 X 和 Y 的十位数单元中的数进行带进位相加，再经过调整，得到 2 位十进制数的相加，结果为 92，其中 9 存放在 DL 中，2 存放在 AL 中。后续的几条指令是将加法相加的结果减去 Z 中的 2 位十进制数。在“SUB AL，Z”指令中，由于 AL 中的内容为 2，而 Z 中的内容为 5，相减后产生借位，在 AAS 调整指令中进行减 6 修正，得到的结果为 7。执行“XCHG AL，DL”指令后，DL=7，AL=9。此时，将 AL 中的十位数与 Z 中的十位数（数值为 6）进行带借位相减，经调整后为 2。所以，表达式 X+Y-Z 的计算结果为 27。其结果的十位数 2 存放在 AH 中，个位数 7 存放在 AL 中。为了将其转换为二进制数，用 AAD 进行调整，十进制数 27 经调整后 AL 中的内容为 1BH。CBW 指令是将 AL 中的符号位扩展到 AH，由于 AL 中的最高位为 0，所以扩展后 AX=001BH，执行“IDIV C”指令后，其中，AL 中存放商，其值为 0DH；AH 中存放余数，其值为 1。并将结果存放在 R 中。该程序的功能为计算表达式 R=（X+Y-Z）/C。根据数据段中给定的数据，程序执行完后 R 中的内容为 AX=010DH。

例 4　现有程序如下：

```
DATA    SEGMENT
ST      DB      ‘PLEASE INPUT X (0…9)：$’
TAB     DW      0，1，8，27，64，125，216，343，512，729
X       DB      ?
Y       DW      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     DX，ST
        MOV     AH，9
        INT     21H
        MOV     AH，1
        INT     21H
        AND     AL，0FH
        MOV     X，AL
        ADD     AL，AL
        MOV     BL，AL
        MOV     BH，0
        MOV     AX，TAB[BX]
        MOV     Y，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）当键盘输入为 6 时，Y= _____。

解：在程序中，首先使用了 DOS 功能调用中的字符串输出功能，将 ST 字符串中的“PLEASE INPUT X（0…9）：”字符显示在屏幕上，然后执行 DOS 中断的键盘输入功能调用，等待操作人员的按键输入。当输入的字符为数字时，将 TAB 表中对应的值送 AX，而 TAB 是以字表示的，所以需形成相应的表的位移量。因而使用“ADD　AL，AL”产生 2 倍的键盘输入值送 BX 作 TAB 表的位移量。在 TAB 表中，由于它依次存放的是 1、2、3、…、9 的立方值，所以该程序的功能是当输入的是数字时，就从表中取出对应的输入值的立方值送 Y。当输入 6 时，Y 的值为 216。

例 5　设 X、Y、Z 为有符号数字变量，请编写程序计算表达式 R=（(X*Y+5）+4*X）/Z 的值。

解：求一个代数式（或表达式）的值，一般的解题方法是根据运算符的优先级顺序进行计算，在计算表达式（(X*Y+5）+4*X）/Z 时，首先计算 X*Y，第二步计算 X*Y+5，第三步计算 4*X，第四步计算（X*Y+5）+4*X，第五步计算（(X*Y+5）+4*X）/Z，其程序流程图如图 4.2 所示。

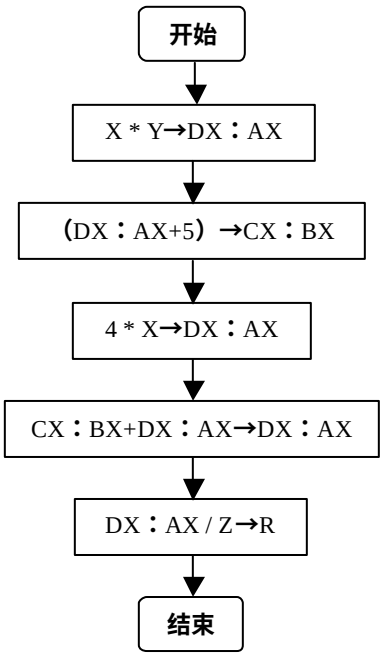


图 4.2　例 5 的程序流程图

在第二步的计算中，要注意 32 位数与常数值的加法运算，它必须用加法和带进位加法才能完成相应的功能。在第四步的计算中，也要注意 32 位数的加法运算。根据程序的流程图，编写的程序清单如下：

```
DATA    SEGMENT
X       DW      ?
Y       DW      ?
Z       DW      ?
R       DW      2  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，X
        IMUL    Y
        ADD     AX，5
        ADC     DX，0
        MOV     CX，DX
        MOV     BX，AX
        MOV     AX，4
        IMUL    Z
        ADD     AX，BX
        ADC     DX，CX
        IDIV    Z
        MOV     R，AX
        MOV     R+2，DX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 6　设 X、Y、Z 为无符号字节变量，编写程序计算表达式 R=（(X+10）*Y/（X+Z）的值。

解：在计算一个代数式（或表达式）的过程中，一般的解题方法是根据运算符的优先级顺序进行计算。在该表达式中，第一步应计算 X+10。在做加法运算时，其相加的结果可能不产生进位，也可能产生进位。因此当结果产生进位而不作处理时，可能会使运算的结果出错，所以在将两个字节相加以后，要将其进位位加到高 8 位上形成 16 位的相加结果。第二步应将 16 位的相加结果与 8 位的 Y 值相乘，而在乘法的运算指令中，两个运算数据的类型必须一致，所以应将 Y 扩展成字，然后才能进行乘法运算实现（X+10）*Y 的计算。第三步应计算 X+Z，第四步应实现（(X+10）*Y/（X+Z）的计算。根据该设计思想，程序流程图如图 4.3 所示。

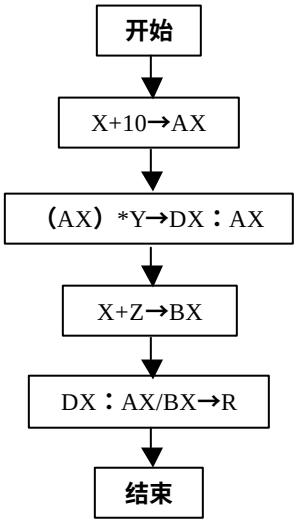


图 4.3　例 6 的程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
X       DB      ?
Y       DB      ?
Z       DB      ?
R       DW      2  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AL, X
        MOV     AH, 0
        ADD     AX, 10
        MOV     BL, Y
        MOV     BH, 0
        MUL     BX
        MOV     CL, X
        MOV     CH, 0
        ADD     CL, Z
        ADC     CH, 0
        DIV     CX
        MOV     R, AX
        MOV     R+2, DX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

4.4 练习题与参考答案

4.4.1 单项选择题

1. 当设计一个程序时，最重要的是（　　）。

A. 程序的结构化　　B. 能使程序正常运行并实现功能
C. 程序的执行速度快　　D. 程序占用的存储空间少

2. 下列描述中正确的是（　　）。

A. 在汇编语言编程中，必须有数据段和代码段
B. 在汇编语言编程中，数据可以安排在代码段
C. 在汇编语言编程中，必须分别完整的定义数据段、代码段、堆栈段和附加段
D. 在汇编语言编程中，必须采用过程的形式编写程序

3. 下列不属于程序设计基本步骤的是（　　）。

A. 提出程序完成的功能、目的、要求。
B. 分析问题，抽象出描述问题的数学模型。
C. 确定解决问题的算法或算法思想。
D. 分配存储空间，工作单元及相应的寄存器。

4. 执行顺序程序时，在程序中（　　）。

A. 有转移指令　　B. 有循环程序段
C. 指令是顺序逐条执行的　　D. 是根据条件分支执行的

5. 在顺序程序的设计中，流程图是非常有用的。关于流程图的作用下列几种说法不正确的是（　　）。

A. 安排程序执行的先后顺序　　B. 明确程序先做什么，后做什么
C. 合理的使用指令　　D. 可以节省内存空间

6. 在设计顺序程序时，要正确使用好两种语句的功能，它们分别是（　　）。

A. 分支语句和循环语句　　B. 指令语句和伪指令语句
C. 顺序语句和非顺序语句　　D. 过程定义语句和控制语句

7. 汇编语言程序设计的流程图是非常有用的，它的每一个执行框表明了整个程序中的某一段程序或某一个功能块，而整个程序有（　　）。

A. 一个入口一个出口　　B. 一个入口多个出口
C. 多个入口一个出口　　D. 多个入口多个出口

8. 在顺序程序设计中，不可能使用的指令是（　　）。

A. 数据传送指令　　B. 算术运算指令
C. 转移指令　　D. 逻辑运算指令

9. 在顺序程序设计的流程图中，不包含（　　）。

A. 开始框　　B. 执行框　　C. 判断框　　D. 结束框

10. 在程序设计过程中，不会影响程序的长度的是（　　）。

A. 确定解决问题的算法　　B. 上机调试
C. 分配存储空间　　D. 编写程序

单项选择题参考答案

1. B	2.B	3.A	4.C	5. D
6. B	7.A	8.C	9.C	10.B

4.4.2　多项选择题

1. 在下列的选项中，不能作为名字的是（　　）。

A. AX　　B. 0ABH　　C. AB　　D. ADD

2. 在下列的选项中，属于指令性语句的有（　　）。

A. MOV　AX，BX　　B. STR　DB　3
C. WORD　PTR　A　　D. IN　AX，80H

3. 在下列指令中，将内存中某一单元的偏移地址送 BX 寄存器的有（　　）。

A. MOV　BX，OFFSET A　　B. MOV　BX，OFFSET [SI]
C. LEA　BX，A　　D. LEA　BX，[SI]

4. 数值表达式的运算符有（　　）。

A. 算术运算符　　B. 逻辑运算符
C. 关系运算符　　D. 属性运算符

5. 地址表达式的运算符有（　　）。

A. 算术运算符　　B. 逻辑运算符
C. 关系运算符　　D. 属性运算符

6. 不能分配内存单元的伪指令（伪操作）有（　　）。

A. 等值语句　　B. 等号语句　　C. 数据定义语句　　D. 段定义语句

7. **段的定义是由下列哪些语句组成的？**

A. SEGNAME SEGMENT　　B. SEGNAME PROC
C. SEGNAME ENDS　　D. SEGNAME ENDP

8. **在段的定义中，如果要将该段定位在以能被** 16 **整除的地址开始，定位方式应选（　　）。**

A. BYTE　　B. WORD　　C. PARA　　D. PAGE

9. **在下列的选项中，属于宏指令的格式有（　　）。**

A. P MICRO
M
ENDM

B. P MICRO A，B
M
ENDM

C. P MICRO
M
P ENDM

D. P MICRO A，B
M
ENDS

10. **在汇编语言程序设计中，上机调试程序时计算机中除了应有** DOS **和编辑器外，还应该有（　　）。**

A. MASM　　B. LINK　　C. DEBUG　　D. WINDOWS

多项选择题参考答案

1. ABD	2. AD	3. ACD	4. ABC	5. ABCD
6. AB	7. AC	8. CD	9. AB	10. ABC

4.4.3 填空题

1. **用汇编语言编写程序，除了要求程序能正常运行和实现预定的功能外，还必须满足程序要结构化，执行的速度要_____，占用的内存空间要_____。**

答：快　少

2. **编写结构化程序的目的是为了使程序简明、_____和_____。**

答：易读　易调试

3. **流程图一般由_____种框组成，在顺序程序的流程图中，一般使用_____框、_____框和_____框。**

答：4**，开始　执行　结束**

4. **无分支，无转移，无循环的程序结构称为_____程序。**

答：顺序

5. 在程序设计过程中，确定解决问题的算法、合理地选择存储空间及工作单元能_____目标程序的长度。

答：缩短

6. 在汇编语言程序中，“MOV　AX，DATA”和“MOV　DS，AX”指令是要将_____送 DS。

答：DATA（数据段）的段地址

7. 对同一问题，如果解决问题的算法不相同，则编写的源程序_____。

答：不相同

8. 在汇编语言源程序中，通常用“MOV　AH，4CH”和“INT　21H”实现 DOS 功能调用的返回，其中的 4CH 为_____号，21H 为_____号。

答：功能　类型

9. 在汇编语言源程序中，“END 表达式”表示_____。而其中的表达式表示程序运行时的_____。

答：整个程序的结束　起始地址

10. 基本的程序设计方法有顺序程序设计、_____、_____和子程序设计四种。

答：分支程序设计　循环程序设计

4.4.4　程序分析题

1. 现有程序如下：

```
DATA    SEGMENT
A       DW      1234H
B       DW      5678H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，A
        XCHG    AX，B
        MOV     A，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序运行后 A 和 B 中的值各为多少？

【解答】（1）将变量 A 和 B 中的内容互换。

（2）（A）=5678H，（B）=1234H。

2. 现有程序如下：

（注：限于篇幅，下列程序中不再列出公共部分，只列出程序中的主要部分。）

```
X     DB      13
Y     DB      5
Z     DB      0
      …
      MOV     AL，X
      ADD     AL，Y
      MOV     CL，2
      SAL     AL，CL
      MOV     Z，AL
```

请回答：（1）该程序完成什么功能？

（2）程序运行完后，Z 中的内容为何值？

（3）若 Y 中的初值为 65H，程序运行完后会出现什么现象？

【解答】（1）计算 Z=（X+Y）*4。

（2）（Z）=48H。

（3）（Z）=38H，运算后的结果会产生溢出。

3. 现有程序如下：

```
BUF1      DB      33H
BUF2      DB      35H
BUF3      DB      0
  …
          MOV     AL，BUF1
          AND     AL，0FH
          MOV     BL，BUF2
          AND     BL，0FH
          MOV     CL，4
          SHL     AL，CL
          OR      AL，BL
          MOV     BUF3，AL
```

请回答：（1）该程序完成什么功能？

（2）程序运行后（BUF3）=_____。

【解答】（1）将 BUF1 和 BUF2 中的非压缩型 BCD 码转换成压缩型 BCD 码并送 BUF3 保存。

（2）（BUF3）=35H。

4. **现有程序如下：**

```
A     DB      56H
B     DB      0
C     DB      0
      …
      MOV     AL，A
      AND     AL，0FH
      ADD     AL，30H
      MOV     B，AL
      MOV     AL，A
      MOV     CL，4
      SHR     AL，CL
      ADD     AL，30H
      MOV     C，AL
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行后** B **和** C **中的内容各为多少？**

【解答】（1）**将** A **中的压缩型** BCD **码转换成非压缩型** BCD **码并送** B **和** C **保存。**

（2）（B）=36H，（C）=35H。

5. **现有程序如下：**

```
TAB       DB      1，2，4，8，10H，20H，40H，80H，90H，0A0H，0C0H
X         DB      ？
Y         DB      0
          …
          MOV     AL，X
          LEA     BX，TAB
          XLAT
          MOV     Y，AL
```

请回答：（1）**该程序完成什么功能？**

（2）**若**（X）=5，**该程序运行完后，**（Y）=_____。

【解答】（1）**根据** X **中的值查** TAB **表，将查表所得值送** Y **保存。**

（2）（Y）=20H。

6. **现有程序如下：**

```
X     DW      100
Y     DW      20
Z     DW      0
      …
      MOV     AX，X
      SUB     AX，Y
      MOV     CL，4
```

```
        SAL     AX，CL
        ADD     AX，20
        SAR     AX，1
        MOV     Z，AX
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行完后** Z **中的内容为多少？**

【解答】（1）**计算表达式** Z=（（X-Y）*16+20）/2。

（2）（Z）=028AH（650）。

7. **现有程序如下：**

```
A       DB      120
B       DB      30
C       DW      0
        …
        MOV     AL，A
        ADD     AL，5
        CBW
        IDIV    B
        MOV     C，AX
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行后** C **中的内容为何值？**

（3）C **中的高低字节各是什么值？**

【解答】（1）**计算表达式** C=（A+5）/B。

（2）（C）=0405H。

（3）**高字节表示余数，低字节表示商。**

8. **现有程序如下：**

```
A       DB      12H
B       DB      56H
C       DB      0
        …
        MOV     AL，A
        ADD     AL，B
        MOV     BL，A
        AND     BL，B
        SAL     BL，1
        SUB     AL，BL
        MOV     C，AL
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行后，**C **中的内容为何值？**

【解答】（1）计算表达式 C=（A+B）-（A　AND　B）*2。

（2）（C）=44H。

9. 现有程序如下：

```
A     DB      10H
B     DB      20H
C     DW      30H
D     DW      2 DUP (0)
      …
      MOV     AL, A
      MUL     B
      ADD     AX, 5
      MOV     BX, C
      SUB     BX, 6
      CWD
      DIV     BX
      MOV     D, AX
      MOV     D+2, DX
```

请回答：（1）该程序完成什么功能？

（2）程序运行后，D 和 D+2 中的内容各为多少？

【解答】（1）计算表达式 D=（A*B+5）/（C-6）。

（2）（D）=0014H，（D+2）=0011H。

10. 现有程序如下：

```
BUF1    DB      12H, 34H
BUF2    DB      35H, 36H
BUF3    DB      2 DUP (0)
        …
        MOV     AL, BUF1+1
        ADD     AL, BUF2+1
        MOV     AH, BUF1
        ADC     AH, BUF2
        MOV     BUF3, AX
```

请回答：（1）该程序完成什么功能？

（2）BUF3 中两个单元中的内容依次是多少？

【解答】（1）将 BUF1 和 BUF2 中的 16 位二进制数相加，其结果送 BUF3 保存。

（2）6AH，47H。

11. 现有程序如下：

```
BUF1    DB      12H, 34H
BUF2    DB      35H, 36H
```

```
BUF3    DB      2  DUP (0)
        …
        MOV     AL，BUF1+1
        ADD     AL，BUF2+1
        DAA
        MOV     BUF3+1，AL
        MOV     AL，BUF1
        ADC     AL，BUF2
        DAA
        MOV     BUF3，AL
```

请回答：（1）**该程序完成什么功能？**

（2）BUF3 **中两个单元中的内容依次是多少？**

【解答】（1）**将** BUF1 **和** BUF2 **中的** 4 **位压缩型** BCD **码相加，其结果送** BUF3 **保存。**

（2）47H，70H。

12. **现有程序如下：**

```
A       DB      35H，37H
B       DB      34H，32H
C       DB      0
        …
        MOV     AL，A
        ADD     AL，B
        AAA
        MOV     BL，AL
        MOV     AL，A+1
        ADC     AL，B+1
        AAA
        MOV     AH，AL
        MOV     AL，BL
        AAD
        MOV     C，AL
```

请回答：（1）**该程序完成什么功能？**

（2）C **中的内容为何值？**

【解答】（1）**将** A、B **中的非压缩型** BCD **码相加，其和转换为二进制数，并送** C **保存。**

（2）63H。

13. **现有程序如下：**

```
BUF1    DB      38H，33H
BUF2    DB      35H，37H
        …
        MOV     AL，A+1
```

```
        SUB     AL，B+1
        AAS
        MOV     DH，AL
        MOV     AL，A
        SBB     AL，B
        AAS
        ADD     AL，30H
        MOV     DL，AL
        MOV     AH，2
        INT     21H
        MOV     DL，DH
        ADD     DL，30H
        MOV     AH，2
        INT     21H
```

请回答：（1）**该程序完成什么功能？**

（2）**显示的结果是什么？**

【解答】（1）**将** BUF1 **和** BUF2 **中的** 2 **位非压缩型** BCD **码相减，并将相减的结果转换成十进制数后再输出。**

（2）**在显示器上显示的是** 2 **和** 6**。**

14. **现有程序如下：**

```
A       DB      39H
B       DB      37H
C       DW      0
        …
        MOV     AL，A
        AND     AL，0FH
        MOV     BL，B
        AND     BL，0FH
        MUL     BL
        AAM
        MOV     BX，AX
        MOV     DL，AH
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        MOV     DL，BL
        ADD     DL，30H
        MOV     AH，2
        INT     21H
```

请回答：（1）**该程序完成什么功能？**

（2）**显示的结果是什么？**

【解答】（1）将 A 和 B 中的非压缩型 BCD 码相乘，其结果转换成十进制数后再输出。

（2）在显示器上显示的是 6 和 3。

15. 现有程序如下：

```
BUF1    DB      100  DUP (?)
BUF2    DB      100  DUP (?)
        …
        MOV     AX, DATA
        MOV     DS, AX
        MOV     ES, AX
        CLD
        LEA     DI, BUF2
        LEA     SI, BUF1
        MOV     CX, 100
        REP     MOVSB
```

请回答：（1）该程序完成什么功能？

（2）在 MOVSB 前可否用其他的重复前缀？

（3）如果程序中没有 CLD 指令行不行？为什么？

【解答】（1）将 BUF1 中的 100 个字节的内容传送到 BUF2 中。

（2）MOVSB 前不能用其他的重复前缀。

（3）不行。如果程序在运行前 DF=0，则可以完成相同的功能。如果程序在运行前 DF=1，则它将以 BUF1 的地址为首地址，然后从此地址开始向低端地址依次传送数据，不能完成相同的功能。

4.4.5 程序填空题

1. 设 X、Y 是无符号数字节变量，下列程序是计算表达式 Z=（X+5）*10/Y 的程序，请在空格处填上适当的语句（不考虑溢出）。

```
DATA    SEGENT
X       DB      ?
Y       DB      ?
Z       DW      0
          (1)
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START： MOV     AX, DATA
          (2)
        MOV     AL, X
        ADD     AL, 5
          (3)
        MUL     BL
```

```
        (4)
        MOV     Z，AX
        MOV     AH，4CH
        (5)
CODE    ENDS
        END     START
```

【解答】(1) DATA　ENDS
(2) MOV　DS，AX
(3) MOV　BL，10
(4) DIV　Y
(5) INT　21H

2. **设 A、B、C 是无符号数字节变量，下列程序是计算表达式** Z=（C+（A+B）*（B+5））/B **的程序段，请在空格处填上适当的指令（不考虑溢出）。**

```
A       DB      ?
B       DB      ?
C       DW      2  DUP (0)
        …
        MOV     AL，A
        ADD     AL，B
        MOV     BL，B
        (1)
        (2)
        ADD     AL，C
        ADC     AH，0
        DIV     B
        MOV     Z，AX
        (3)
```

【解答】(1) ADD　BL，5
(2) MUL　BL
(3) MOV　Z+2，DX

3. **设 X、Y 是有符号数字变量，下列程序是计算表达式** S=（(X+5) *Y+4）/（X－Y）**的程序段，请在程序的空格处填上适当的指令（不考虑溢出）。**

```
X       DW      ?
Y       DW      ?
S       DW      2  DUP (0)
        …
        MOV     AX，X
        ADD     AX，5
        IMUL    Y
        (1)
```

```
        (2)
MOV     BX，X
SUB     BX，Y
        (3)
MOV     S+2，AX
        (4)
```

【解答】（1）ADD　AX，4
（2）ADC　DX，0
（3）IDIV　BX
（4）MOV　S，DX

4. **下列程序段是将** BUF **缓冲区中的两个字节的非压缩型** BCD **码转换成** ASCII **码并送显示器显示的程序，请在程序的空格处填上适当的指令。**

```
BUF     DB      5，7
        …
        MOV     DL，BUF
        (1)
        MOV     AH，2
        (2)
        (3)
        ADD     DL，30H
        MOV     AH，2
        (4)
```

【解答】（1）ADD　DL，30H
（2）INT　21H
（3）MOV　DL，BUF+1
（4）INT　21H

5. **下列程序段是将** A **和** B **中的** 24 **位的无符号数相加，其结果存放在双字的** C **变量中的程序，请在程序段的空格处填上适当的指令。**

```
A       DB      3  DUP (?)
B       DB      3  DUP (?)
C       DD      0
        …
        MOV     AL，A
        (1)
        XCHG    CH，CL
        MOV     BL，B
        (2)
        XCHG    DH，DL
        ADD     CX，DX
        (3)
```

```
        MOV     AH，0
        ____(4)____
        ADC     AH，BH
        MOV     C，CX
        ____(5)____
```

【解答】(1) MOV　CX，WORD PTR　A+1
(2) MOV　DX，WORD PTR　B+1
(3) ADC　AL，BL
(4) MOV　BH，0
(5) MOV　C+2，AX

6. **下列程序段是将 A 和 B 中两个数字字符串（非压缩型 BCD 码）相加，其结果存放在 C 中的程序，请在程序段的空格处填上适当的指令。**

```
A       DB      ‘35’
B       DB      ‘92’
C       DB      3 DUP (0)
        …
        MOV     AL，A
        MOV     BL，B
        ____(1)____
        AAA
        ____(2)____
        MOV     AL，A+1
        MOV     BL，B+1
        ____(3)____
        AAA
        MOV     C+1，AL
        MOV     AH，0
        ____(4)____
        MOV     C，AH
```

【解答】(1) ADD　AL，BL
(2) MOV　C+2，AL
(3) ADC　AL，BL
(4) ADC　AH，0

7. **下列程序段是将 X 和 Y 中的 32 位的无符号的二进制数相乘，其结果存放在 Z 中的程序，请在程序段的空格处填上适当的指令。**

```
X       DW      2 DUP (?)
Y       DW      2 DUP (?)
Z       DW      4 DUP (0)
        …
        MOV     AX，Y+2
        ____(1)____
```

```
        MOV     Z+6，AX
        MOV     Z+4，DX
        MOV     AX，Y+2
        (2)
        ADD     Z+4，AX
        ADC     Z+2，DX
        ADC     Z，0
        MOV     AX，Y
        (3)
        ADD     Z+4，AX
        ADC     Z+2，DX
        ADC     Z，0
        MOV     AX，Y
        (4)
        ADD     Z+2，AX
        ADC     Z，DX
```

【解答】（1）MUL　　X+2

（2）MUL　　X

（3）MUL　　X+2

（4）MUL　　X

8. **下列程序段是将 X 中的 64 位的无符号的二进制数除以 Y 中的 16 位的无符号的二进制数，其结果的商存放在 W，余数存放在 N 中的程序，请在程序段的空格处填上适当的指令。**

```
X       DW      4 DUP (?)
Y       DW      ?
W       DW      4 DUP (0)
N       DW      0
        …
        MOV     DX，0
        MOV     AX，X+6
        (1)
        DIV     BX
        MOV     W+6，AX
        MOV     AX，X+4
        DIV     BX
        (2)
        MOV     AX，X+2
        DIV     BX
        (3)
        MOV     AX，X
        DIV     BX
        (4)
        (5)
```

【解答】(1) MOV　　BX，Y
(2) MOV　　W+4，AX
(3) MOV　　W+2，AX
(4) MOV　　W，AX
(5) MOV　　N，DX

9. **下列程序段是利用宏定义和 DOS 功能调用将 ST 中的字符串显示在显示器上的程序，请在程序段的空格处填上适当的指令或参数。**

```
DISPLAY     MACRO   X, Y
            MOV     AH, X
                (1)
            INT     21H
                (2)
DATA        SEGMENT
ST          DB      'ABCD$'
DATA        ENDS
            …
            DISPLAY  (3), (4)
```

【解答】(1) LEA DX，U
(2) ENDM
(3) 9
(4) ST

10. **下列程序段是将 BUF1 字缓冲区中的内容送 300H 输出端口，将 200H 端口中的内容送 BUF2 字缓冲区的程序，请在程序段的空格处填上适当的指令。**

```
BUF1    DW      ?
BUF2    DB      ?
        …
        LEA     BX, BUF1
        MOV     AX, [BX]
            (1)
            (2)
            (3)
            (4)
        MOV     BUF2, AL
```

【解答】(1) MOV　　DX，300H
(2) OUT　　DX，AX
(3) MOV　　DX，200H
(4) IN　　AL，DX

4.4.6 程序设计题

1. 设 X、Y、Z 为无符号的 16 位二进制数，编写程序计算表达式 W=（X+8）*Z+（X*Y）（不考虑加法的溢出）。

【解答】
```
DATA    SEGMENT
X       DW      ?
Y       DW      ?
Z       DW      ?
W       DW      2  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，X
        ADD     AX，8
        MUL     Z
        MOV     BX，AX
        MOV     CX，DX
        MOV     AX，X
        MUL     Y
        ADD     AX，BX
        ADC     DX，CX
        MOV     W，AX
        MOV     W+2，DX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

2. 设 X、Y、Z 为有符号的 16 位二进制数，编写程序计算表达式 W=(X+Y+10)*Z/(X+Y)（不考虑加法的溢出），其中 W 存放商，W+2 存放余数。

【解答】
```
DATA    SEGMENT
X       DW      ?
Y       DW      ?
Z       DW      ?
W       DW      2  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，X
```

```
        ADD     AX, Y
        ADD     AX, 10
        IMUL    Z
        MOV     BX, X
        ADD     BX, Y
        IDIV    BX
        MOV     W, AX
        MOV     W+2, DX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

3. **编一程序从键盘输入一个字母，然后找出它的前导字符和后续字符（如 B 的前导字符为 A，后续字符为 C），并顺序显示输出这三个字符。**

【解答】

```
DATA    SEGMENT
ST      DB      4 DUP ('$')
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AH, 1
        INT     21H
        MOV     ST+1, AL
        DEC     AL
        MOV     ST, AL
        ADD     AL, 2
        MOV     ST+2, AL
        LEA     DX, ST
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

4. **将 BUF 中的 16 位数分成四组，每组四位，然后把这四组数分别存放在 A、B、C、D 字节变量中。**

【解答】

```
DATA    SEGMENT
BUF     DW      ?
A       DB      0
B       DB      0
C       DB      0
```

```
D       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，BUF
        AND     AL，0FH
        MOV     D，AL
        AND     AH，0FH
        MOV     B，AH
        MOV     AX，BUF
        MOV     CL，4
        SHR     AL，CL
        MOV     C，AL
        MOV     CL，4
        SHR     AH，CL
        MOV     A，AH
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

5. **将** BUF **中的** 8 **位二进制数转换成八进制数，其中的最高位存放在** A **字节单元中，最低位存放在** C **字节单元中。同时显示转换后的结果。**

【解答】

```
DATA    SEGMENT
BUF     DB      ？
A       DB      0
B       DB      0
C       DB      0
D       DB      4  DUP（‘ $ ’）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，BUF
        AND     AL，07H
        MOV     C，AL
        ADD     AL，30H
        MOV     D+2，AL
        MOV     AL，BUF
        MOV     CL，3
        SHR     AL，CL
```

```
        AND     AL, 07H
        MOV     B, AL
        ADD     AL, 30H
        MOV     D+1, AL
        MOV     AL, BUF
        MOV     CL, 6
        SHR     AL, CL
        MOV     A, AL
        ADD     AL, 30H
        MOV     D, A
        LEA     DX, D
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

6. **将 A 和 B 中的 2 位压缩型 BCD 码相加后并输出结果。**

【解答】

```
DATA    SEGMENT
A       DB      35
B       DB      63
C       DB      4 DUP ('$')
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AL, A
        ADD     AL, B
        DAA
        MOV     AH, 0
        ADC     AH, 30H
        MOV     C, AH
        MOV     AH, AL
        MOV     CL, 4
        SHR     AH, CL
        ADD     AH, 30H
        MOV     C+1, AH
        AND     AL, 0FH
        ADD     AL, 30H
        MOV     C+2, AL
        LEA     DX, C
        MOV     AH, 9
```

```
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

7. **将 A 中的 2 位非压缩型 BCD 码与 B 中的 2 位非压缩型 BCD 码相乘，并将结果存放在 C 字节变量中。**

【解答】

```
DATA    SEGMENT
A       DB      3，5
B       DB      4，8
C       DB      4  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        MUL     B
        AAM
        MOV     BX，AX
        MOV     AL，A
        MUL     B+1
        AAM
        ADD     BH，AL
        ADC     AH，0
        MOV     CL，AH
        MOV     AL，A+1
        MUL     B
        AAM
        ADD     BH，AL
        ADC     CL，AH
        MOV     CH，0
        ADC     CH，0
        MOV     AL，A+1
        MUL     B+1
        AAM
        ADD     AL，CL
        ADC     AH，CH
        MOV     C，BX
        MOV     C+2，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

8. **将 A 字节变量中的 48 位无符号数与 B 字节变量中的 16 位无符号数相乘，并将结果存放在 C 字节变量中。**

【解答】
```
DATA    SEGMENT
A       DW      3  DUP (?)
B       DW      ?
C       DW      4  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，A+4
        MUL     B
        MOV     C+6，AX
        MOV     C+4，DX
        MOV     AX，A+2
        MUL     B
        ADD     C+4，AX
        ADC     DX，0
        MOV     C+2，DX
        MOV     AX，A
        MUL     B
        ADD     C+2，AX
        ADC     DX，0
        MOV     C，DX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

9. **用宏定义、宏调用实现 A 和 B 变量中的 2 位非压缩型 BCD 码相加，然后显示输出相加的结果。**

【解答】
```
M       MACRO   X，Y
        MOV     AL，X
        ADC     AL，Y
        AAA
        ENDM
N       MACRO   W
        MOV     DL，W
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        ENDM
```

```
DATA    SEGMENT
A       DB      6，7
B       DB      8，8
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        CLD
        M       A+1，B+1
        MOV     CL，AL
        M       A，B
        MOV     CH，AL
        M       0，0
        N       AL
        N       CH
        N       CL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

10. **设** TAB **是存放数字开方后的数字表，其中每个开方值占两个字节，低字节存放开方的值的整数部分，高字节存放余数部分，试用查表的方法求出** X **字单元中的开方值，并将结果存放在** Y **单元中。**

【解答】

```
DATA    SEGMENT
TAB     DW      1000  DUP（?）
X       DW      ?
Y       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     BX，TAB
        MOV     AX，X
        SHL     AX，1
        ADD     BX，AX
        MOV     AX，[BX]
        MOV     Y，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

第 5 章　分支程序设计

分支程序是一种基本的和重要的程序结构形式。分支程序设计是程序设计中常用的一种设计方法。学习分支程序的设计方法，有助于在程序设计过程中，根据不同的情况，分别进行各种不同的处理。

在本章的学习中，要求掌握分支程序的结构特点和结构形式；熟练地掌握双分支程序设计中条件的产生与判断的程序设计方法和技巧；理解和掌握转移表法分支程序设计、地址表法分支程序设计、逻辑分解法分支程序设计等基本原理、设计方法和技巧。

【学习重点】

双分支程序设计方法。

【学习难点】

三种分支程序的设计的方法和技巧。

5.2.1　知识体系结构

本章中所讲述的知识体系结构如图 5.1 所示。

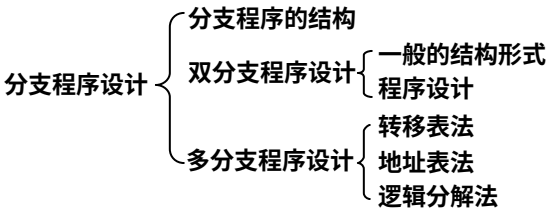


图 5.1　分支程序设计知识体系结构

5.2.2　知识点与考核要求

1. 分支程序的概念和结构，应达到“理解”的层次。

（1）什么是分支程序。了解双分支程序和多分支程序的结构形式及其在程序设计中的不同作用。

（2）双分支程序和多分支程序的程序流程图的特点和结构形式。

2. 双分支程序设计，应达到“综合应用”的层次。

（1）双分支程序的设计方法和技巧，重点在产生分支的程序段。

（2）灵活使用产生条件指令和判断指令，根据产生的条件判断程序的转移，掌握双分支程序的设计方法和技巧。

（3）能够分析双分支程序，能够应用双分支程序设计方法编写程序。

3. 多分支程序设计，应达到“简单应用”的层次。

（1）转移表法和地址表法多分支程序的设计原理，转移表和地址表的结构和作用，在两种方法中实现多分支的设计方法和技巧。

（2）逻辑分解法多分支程序的设计原理，关键字的组成和作用，逻辑分解法实现多分支的设计方法和技巧。

5.3 例题分析

例 1　现有程序如下：

```
DATA        SEGMRNT
STR         DB        ‘PLEASE  INPUT  X (0~9)：$’
ER          DB        ‘INPUT  ERROR！$’
TAB         DW        0, 1, 8, 27, 64, 125, 216, 343, 512, 729
X           DB        0
Y           DW        0
DATA        ENDS
CODE        SEGMENT
            ASSUME    CS：CODE, DS：DATA
START：     MOV       AX, DATA
            MOV       DS, AX
            LEA       DX, STR
            MOV       AH, 9
            INT       21H
            MOV       AH, 1
            INT       21H
            CMP       AL, 30H
            JB        N
```

```
            CMP     AL，39H
            JA      N
            AND     AL，0FH
            MOV     X，AL
            ADD     AL，AL
            MOV     BL，AL
            MOV     BH，0
            MOV     AX，TAB[BX]
            MOV     Y，AX
EXIT：      MOV     AH，4CH
            INT     21H
N：         LEA     DX，ER
            MOV     AH，9
            INT     21H
            JMP     EXIT
CODE        ENDS
            END     START
```

请回答：该程序完成什么功能？

解：在该程序中，首先利用 DOS 功能调用显示字符串“ PLEASE　INPUT　X（0~9）：”，然后再进行 DOS 功能调用等待键盘输入。当输入的字符为非数字时，就利用 DOS 功能调用显示“ INPUT　ERROR !”字符串；当输入的字符为数字时，就计算输入数字的立方值并送 Y 保存。在计算立方值的过程中，由于其数字的立方值存放在 TAB 表中，而 TAB 表的每个数字占用两个字节单元，为了使输入的数字与 TAB 表中的立方值字单元地址对应，需将输入的数字乘 2（即用“ ADD　AL，AL ”指令）。然后以 TAB 表的首地址，以键盘输入值的 2 倍值为位移量，从该字单元中取出立方值送 Y。其流程图如图 5.2 所示。

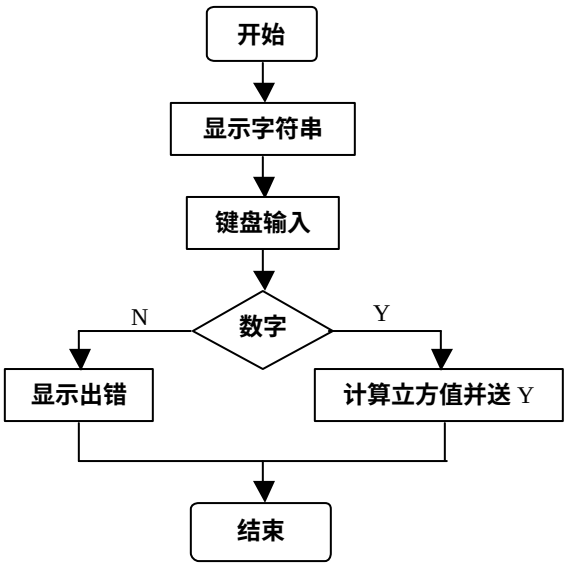


图 5.2　例 1 程序流程图

例 2　现有程序如下：

```
DATA        SEGMENT
X           DW      ?
Y           DW      ?
Z           DW      ?
MAX         DW      0
DATA        ENDS
CODE        SEGMENT
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
            MOV     AX，X
            CMP     AX，Y
            JG      XZ
            MOV     AX，Y
            CMP     AX，Z
            JG      M
ZM：        MOV     AX，Z
            JMP     M
XZ：        CMP     AX，Z
            JNG     ZM
M：         MOV     MAX，AX
            MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

请回答：1. **该程序完成什么功能？**

2. **若将** JG **和** JNG **分别改为** JA **和** JNB**，程序有何不同？**

解：该程序是将数据段中的三个数 X、Y **和** Z **进行两两比较，先取两个数进行比较，较大数送** AX **寄存器。然后再和另外一个数进行比较，最后将最大数送给** MAX **变量保存。该程序实现了将** X、Y、Z **中的最大的有符号数送** MAX **的功能。程序流程图如图** 5.3 **所示。**

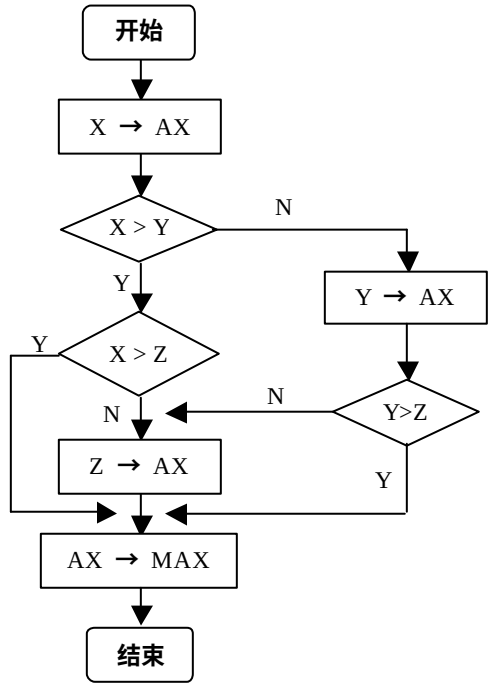


图 5.3　例 2 程序流程图

例 3　下列程序是计算表达式

$$W=\begin{cases}1 & X\geq 0，Y\geq 0\\ -1 & X<0，Y<0\\ 0 & X、Y\text{ 为异号}\end{cases}$$

的程序。请在程序中的空格处填写适当的指令（X，Y 为字节数据）。

```
DATA        SEGMENT
X           DB      ?
Y           DB      ?
W           DB      ?
DATA        ENDS
CODE        SEGMENT
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
                  (1)
            JS      L1
            CMP     Y，0
                  (2)
            MOV     W，1
            JMP     EXIT
L1：        CMP     Y，0
```

```
                    (3)
                MOV     W, -1
                    (4)
L2：            MOV     W, 0
EXIT：          MOV     A, 4CH
                INT     21H
CODE            ENDS
                END     START
```

解：要实现该程序的功能，必须对参加运算的数据 X 和 Y 进行判断。当两个操作数都为正数时，应将 W 置 1；当两个操作数都为负数时，则应将 W 置为-1；当两个操作数为异号时，则应将 W 清 0。根据此要求，其程序的流程图如图 5.4 所示。

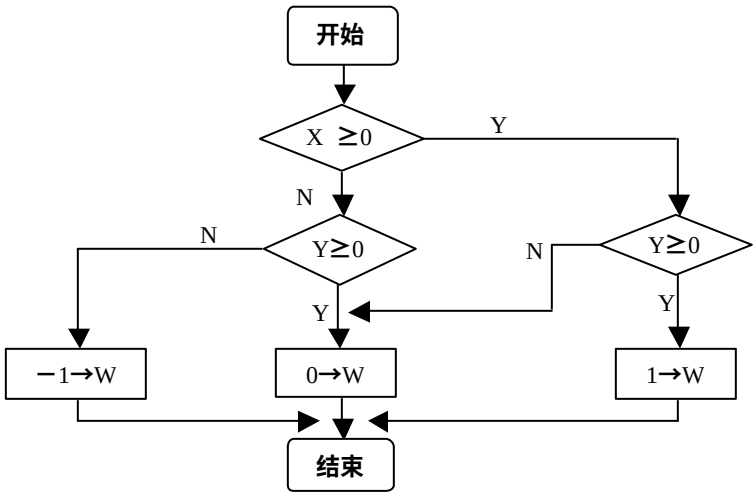


图 5.4　例 3 的程序流程图

由此，在比较判断转移中，第（1）空格应填“ CMP　X，0 ”，第（2）空格应填“ JL　L2 ”，第（3）空格应填“ JGE　L2 ”，第（4）空格应填“ JMP　EXIT ”。

例 4　下列程序是两个字符串的比较，若两个字符串相等，则输出“ YES ”；若不相等，则输出“ NO ”的程序。请在程序的空格处填上适当的指令。

```
DATA        SEGMENT
STR1        DB      ‘I  AM  A  STUDENT.’
C1          =       $-STR1
STR2        DB      ‘I  AM  A  STUDENT!’
C2          =       $-STR2
Y           DB      ‘YES$’
N           DB      ‘NO$’
DATA        ENDS
CODE        SEGENT
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
```

```
            (1)
            CLD
            MOV     CX，C1
            (2)
            JNZ     NOEQ
            LEA     SI，STR1
            (3)
            REPE    CMPSB
            (4)
            MOV     AH，9
            LEA     DX，Y
            INT     21H
            JMP     NEXT
NOEQ：      MOV     AH，9
            MOV     DX，N
            INT     21H
NEXT：      MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

解：要比较两个字符串是否相等，要从两方面进行考虑。一是要比较两个字符串的个数是否相等，若不相等则表示两个字符串不相等。只有当两个字符串的字符个数相等的情况下才有可能相等。二是当两个字符串个数相等之后，再来比较字符串中的内容是否相等。若不相等，则表示字符串还是不完全相等，则应显示输出“ NO **”；若相等，则应显示输出“** YES **”。本例题的程序流程图如图** 5.5 **所示。**

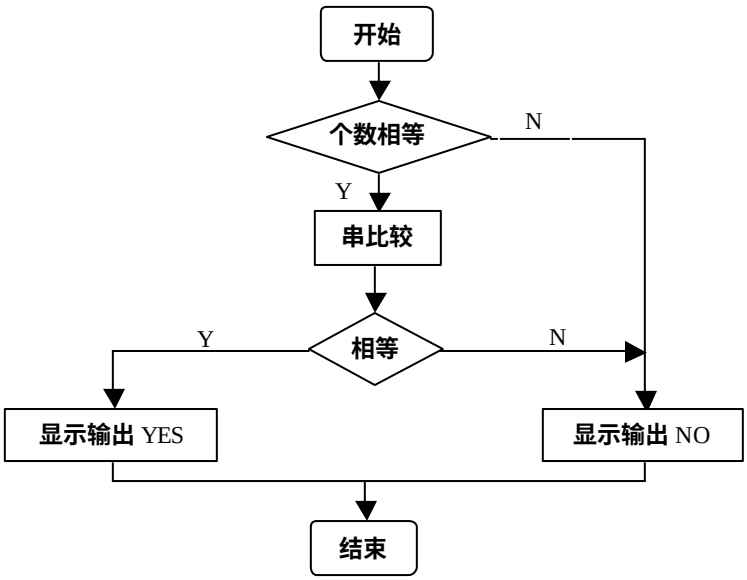


图 5.5　**例** 4 **的程序流程图**

由于在程序中使用的是串操作指令，应将源串和目的串的段地址分别送 DS 和 ES。所以第（1）空格应填“MOV　ES，AX”。在比较字符串个数的过程中，因字符串 STR1 的个数 C1 已送 CX，则应将 CX 中的内容与字符串 STR2 的个数 C2 进行比较，所以在第（2）空格应填“CMP　CX，C2”。在进行串比较之前，应将串的首地址和个数分别送对应的寄存器。当比较两个串的字符个数时，串的字符个数已送 CX，只有当两个串的字符个数相等时才进行串比较。所以可用 CX 中的内容作比较的次数。而 STR1 的首地址已送 SI，所以在第（3）空格应填“LEA　DI，STR2”。在比较完之后有两种情况可以退出比较。一是个数比较完，二是比较过程中有不相等的字符。判断比较后的两个字符串是否相等，应根据退出串比较后的零标志位 ZF 来判断，而不是根据 CX 的个数来判断。这是因为在比较最后一个字符时，也有可能其个数为 0，而字符有可能不相等的情况存在。所以在第（4）空格应填“JNZ　NOEQ”。

例 5　设 X 为有符号的字节变量，编写程序计算表达式

$$Y=\begin{cases} X+4 & X\geq 0 \\ X-10 & X<0 \end{cases}$$

解：在计算的过程中，需根据 X 的不同值进行计算。当 X 的值大于等于 0 时，需求出 X＋4，并将其结果送 Y；当 X 的值小于零时，需求出 X-10，并将其结果送 Y。在计算表达式的过程中，当 X 大于等于 0 时是进行加法运算，计算的结果有可能会超出一个字节的表示范围。为了保证结果的正确性，需将一个字节的内容转换成一个字相加，这样才能保证字节数相加后产生的进位不会被丢失。同理，在减法的过程中，需将一个字节的内容转换成一个字相减。程序流程图如图 5.6 所示。

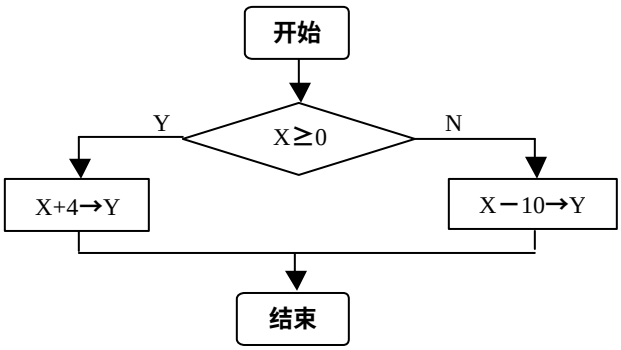


图 5.6　例 5 的程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
X       DB      ?
Y       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
```

```
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AL，X
         CBW
         CMP     AX，0
         JS      M
         ADD     AX，4
         MOV     Y，AX
         JMP     EXIT
M：      SUB     AX，10
         MOV     Y，AX
EXIT：   MOV     AH，4CH
         INT     21H
CODE     ENDS
         END     START
```

例 6　设 X，Y 是一个无符号的字节变量，试编写程序计算表达式

$$W=\begin{cases}10 & X=0\\ Y+2 & X=1\\ Y*4 & X=2\\ Y+10 & X=3\\ Y*2+3 & X\text{ 为其他值}\end{cases}$$

解：要编写计算此表达式的程序，可用逻辑分解法、地址表法和转移表法等方法设计该程序。在本例中只列出运用地址表法设计的程序。在程序设计过程中，首先在数据段中定义一个程序转移地址的地址表变量，这样，在程序中就能根据 X 的不同值转移到由地址表指示的不同程序段中。在计算地址表的过程中，由于程序转移的地址是 16 位的偏移量，需占用两个字节的存储空间。因此，当 X=0 时，占用该表的 0、1 两个字节；当 X=1 时，占用 2、3 两个字节；X=2 时，占用 4、5 两个字节；X=3 时，占用 6、7 两个字节，X 为其他值时占用 8、9 两个字节。由此可知，当 X＜4 时将 X 的值乘以 2 再取其低 4 位的值就是该入口地址所在表的位移量，当 X≥4 时，其入口地址的位移量都是 8。所以在程序设计中应该对此进行区分。根据此分析，程序流程图如图 5.7 所示。

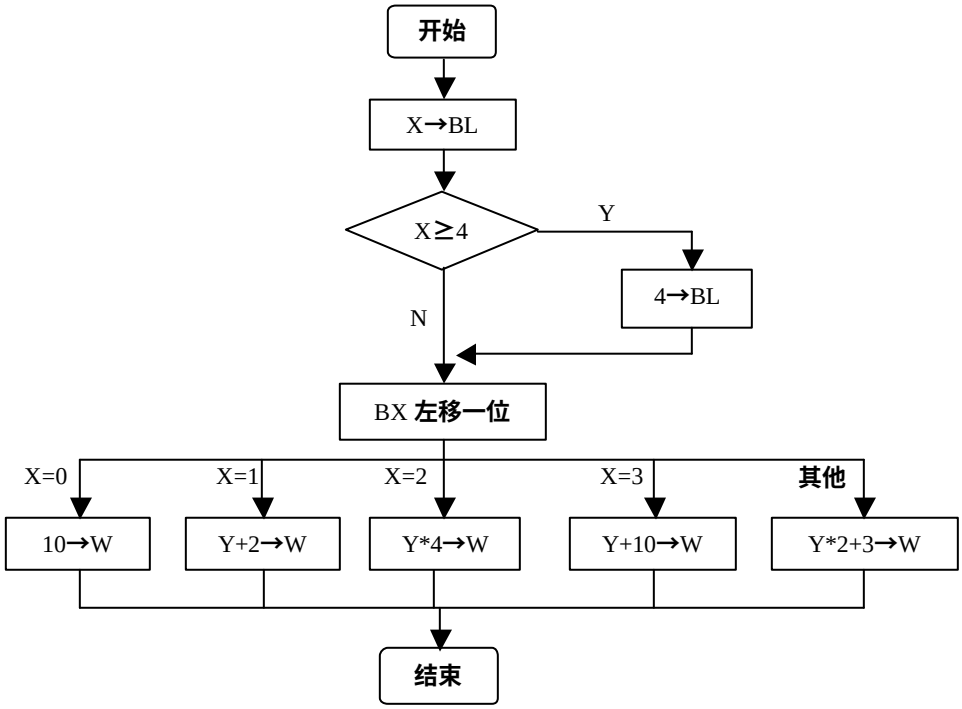


图 5.7　**例** 6 **程序流程图**

编写的程序清单如下：

```
DATA    SEGMENT
TAB     DW      X0, X1, X2, X3, XX
X       DB      ?
Y       DB      ?
W       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     BH, 0
        MOV     BL, X
        CMP     BL, 4
        JB      L
        MOV     BL, 4
L：      SHL     BL, 1
        JMP     TAB[X]
X0：     MOV     W, 10
        JMP     EXIT
X1：     MOV     AL, Y
```

```
        MOV     AH，0
        ADD     AX，2
        MOV     W，AX
        JMP     EXIT
X2：    MOV     AL，Y
        MOV     BL，4
        MUL     BL
        MOV     W，AX
        JMP     EXIT
X3：    MOV     AL，Y
        MOV     AH，0
        ADD     AX，10
        MOV     W，AX
        JMP     EXIT
X4：    MOV     AL，Y
        MOV     AH，0
        SHL     AX，1
        ADD     AX，3
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 7　设 X、Y、Z 为无符号字的节变量，编写一个程序，根据三个数的比较显示如下信息：

（1）如果三个数都不相等则显示 0。
（2）如果三个数中有两个相等则显示 1。
（3）如果三个数全相等则显示 2。

解：要比较三个数是否完全相等，则应进行两两比较，当 A＝B，A＝C 时，表示三个数都相等；当 A＝B、A≠C 或 A≠B、B＝C 或 A＝C、B≠C 时表示两个数相等；当 A≠B、B≠C、A≠C 时表示三个数都不等。则程序流程图如图 5.8 所示。

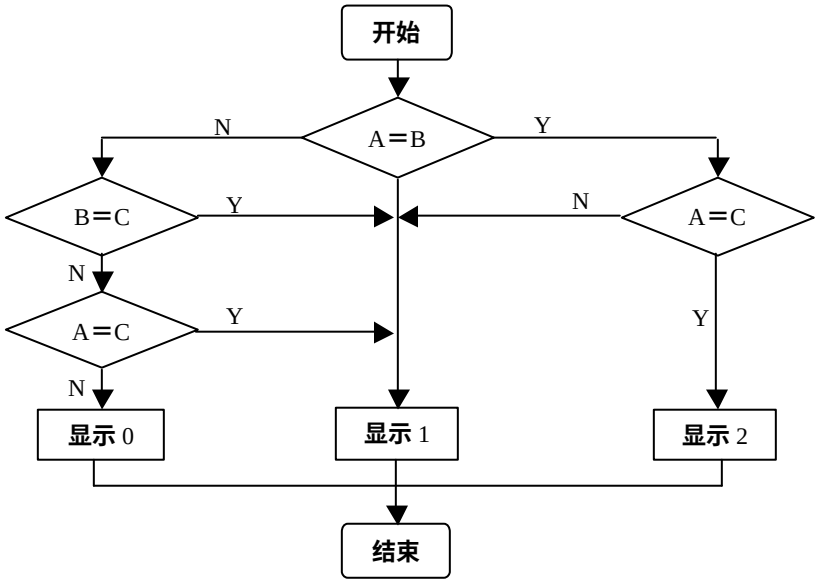


图 5.8　例 7 的程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
A       DB      ?
B       DB      ?
C       DB      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        MOV     BL，B
        MOV     CL，C
        MOV     DL，0
        CMP     AL，BL、
        JNE     L
        INC     DL
        CMP     AL，CL
        JNE     DISP
        INC     DL
        JMP     DISP
L：      CMP     BL，CL
        JNE     M
        INC     DL
        JMP     DISP
M：      CMP     AL，CL
```

```
        JNE     DISP
        INC     DL
DISP：  ADD     DL，30H
        MOV     AH，2
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 8　试编写一程序，将键盘输入的字符送 X 并进行判断。若输入的字符为大写字母或数字，则原样显示；若为小写字母则转换为大写字母后再显示；若为其他字符则显示“＄”符号。

解：根据题意，该程序应分三步完成，第一步键盘输入。第二步判断键盘输入的字符，根据字符的不同作不同的处理。第三步显示输出。其程序流程图如图 5.9 所示。

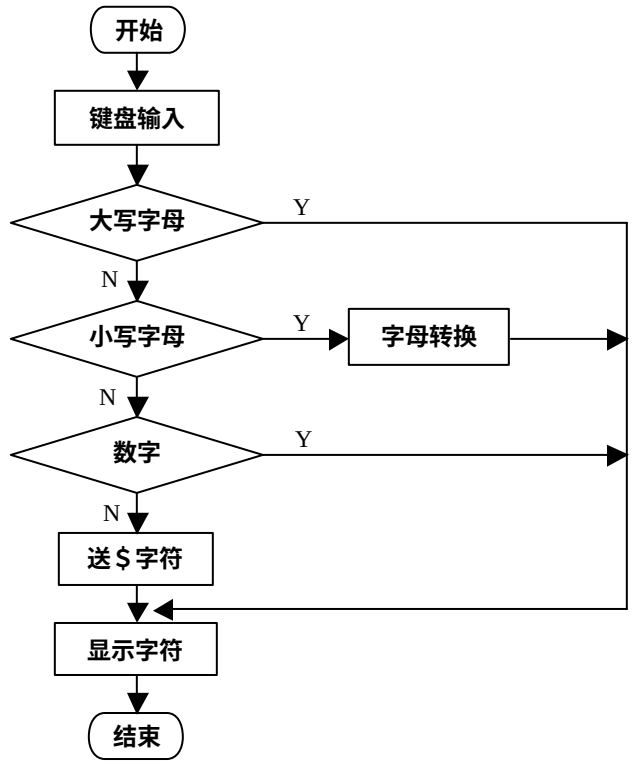


图 5.9　例 8 的程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
X       DB      0
DATA    ENDS
CODE    SEGMENT
```

```
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AH，1
        INT     21H
        MOV     X，AL
        CMP     AL，‘A’
        JB      L
        CMP     AL，‘Z’
        JA      L
        MOV     DL，AL
        JMP     DISP
L：     CMP     AL，‘a’
        JB      M
        CMP     AL，‘z’
        JA      M
        SUB     AL，20H
        MOV     DL，AL
        JMP     DISP
M：     CMP     AL，30H
        JB      N
        CMP     AL，39H
        JA      N
        MOV     DL，AL
        JMP     DISP
N：     MOV     DL，‘$’
DISP：  MOV     AL，2
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 9　在一个从小到大顺序排列的无符号数字数组 ARY 中，数组的第一个单元存放着数组长度。在字变量 X 中有一个无符号数，要求在数组中查找（X），如果找到则使 CF=0，并在 Y 字变量中给出该元素在数组中的偏移地址；否则，则使 CF=1。

解：如果要从数组中查找某一个数，可以使用顺序查找的方法。由于已知数组中的数是有序存储，所以可以采用折半查找法以提高查找效率。折半查找法是先取有序数组的中间元素与查找值相比较，如果相等则查找成功；如果查找值大于中点元素，则再取高半部的中间元素与查找值相比较；如果查找值小于中间元素，则再取低半部的中间元素与查找值相比较；如此重复直到查找成功或最终未找到该数（查找不成功）为止。折半查找法的效率高于顺序查找法，对于长度为 n 的数组，顺序查找法的平均比较次数为 n/2，而折半查

找法的平均比较次数为 $\log_2 n$。所以，如果数组长度为 100，则顺序查找法平均要作 50 次比较，而折半查找法平均作 7 次比较就可以了。

在一个长度为 n 的有序数组 r 中，设查找元素为 k，其折半查找算法可描述如下：

① 初始化被查找数组的首、尾下标，low←1，high←n。

② 若 low > high，则查找失败，置 CF=l，退出程序；
否则，计算中点元素：mid←（low + high）/2。

③ k 与中点元素 r［mid］比较。若 k=r[mid]，则查找成功，程序结束；
若 k<r［mid］，则转步骤④；若 k>r[mid]，则转步骤⑤。

④ 低半部分查找（lower），high←mid-1，返回步骤②，继续查找。

⑤ 高半部分查找（higher），low←mid+l，返回步骤②，继续查找。

在程序的设计过程中，首先要把查找值与数组的第一个元素和最后一个元素相比较，如果相等，则表示找到，结束查找；如果小于第一个元素或大于最后一个元素则表示未找到，结束查找；否则，表示查找值是大于第一个元素和小于最后一个元素，因此查找时应该从第一个元素开始折半查找。采用折半查找算法的程序流程图如图 5.10 所示。

编写的程序清单如下：

```
DATA    SEGMENT
ARY     DW      12，5，7，8，9，12，13，14，16，18，34，56，344
X       DW      12
Y       DW      0
LOWIDX  DW      0
HIGHIDX DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，X           ；与第一个元素比较
        CMP     AX，ARY+2
        JA      CHK             ；大于第一个元素时，转移到 CHK
        LEA     SI，ARY+2
        JE      EXIT            ；等于第一个元素时，转移到 EXIT
        STC                     ；不等于第一个元素时，置 CF=1
        JMP     EXIT
CHK：    MOV     SI，ARY         ；指向最后一个元素
```

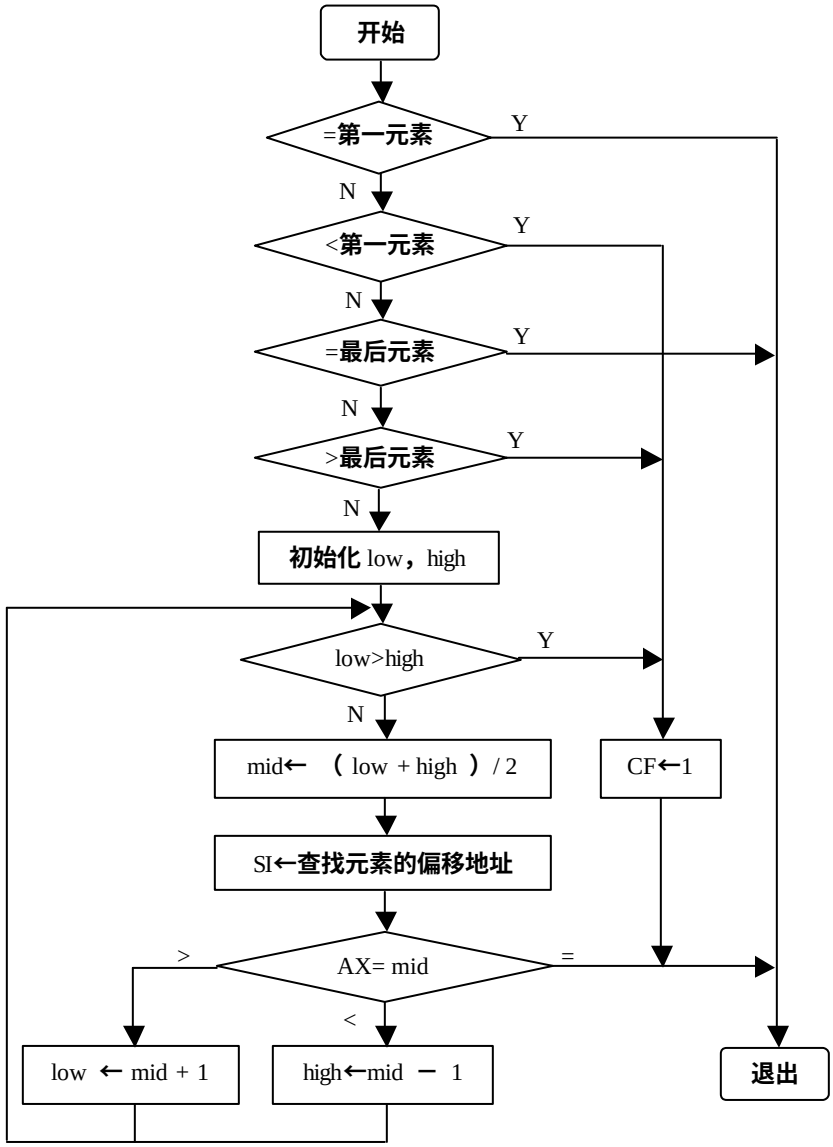


图 5.10　**折半查找算法的程序流程图**

```
        SHL     SI，1
        ADD     SI，OFFSET ARY
        CMP     AX，[SI]          ；与最后一个元素比较
        JB      SEARCH            ；小于最后一个元素时，转移到 SEAR
        JE      EXIT              ；等于最后一个元素时，转移到 EXIT
        STC                       ；不等于最后一个元素时，置 CF=1
        JMP     EXIT
SEARCH：MOV     LOWIDX，1         ；置首、尾下标
        MOV     BX，ARY
        MOV     HIGHIDX，BX
        LEA     BX，ARY           ；送数组的首地址
```

```
MID：       MOV     CX，LOWIDX
            MOV     DX，HIGHIDX
            CMP     CX，DX          ；查找完，则转移到 NOMATCH
            JA      NOMATCH
            ADD     CX，DX          ；未完，则计算中点元素的下标
            SHR     CX，1
            MOV     SI，CX
            SHL     SI，1
            CMP     AX，[BX+SI]     ；比较是否相等
            JZ      EXIT            ；相等，表示找到，转移到 EXIT
            JA      HIGHER          ；大于，则需往高端查找
            DEC     CX
            MOV     HIGHIDX，CX     ；求出高端下标
            JMP     MID
HIGHER：    INC     CX
            MOV     LOWIDX，CX      ；求出低端下标
            JMP     MID
NOMATCH：   STC                     ；送未找到标志
EXIT：      MOV     Y，SI
            MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

5.4 练习题与参考答案

5.4.1 单项选择题

1. **双分支程序设计中实现分支的指令是（　　）指令。**

A. **条件转移**　　B. **无条件转移**　　C. **移位**　　D. **算术运算**

2. **条件判断转移是根据标志寄存器中的标志位来判断的，可以作为条件判断转移标志位的共有（　　）位。**

A. 4　　B. 5　　C. 6　　D. 9

3. **条件转移指令的转移范围是（　　）。**

A. 0～65535　　B. -32768～32767　　C. 0～255　　D. -128～127

4. **用一条条件转移指令一次可实现（　　）个分支。**

A. 2　　B. 3　　C. 4　　D. **多**

5. 下列指令会影响标志位的是（　　）。

A. JMP　L　　B. JC　L　　C. MOV　AL，L　　D. SHL　AL，1

6. 当两个无符号数进行相加时，执行“JC　L”指令表示若（　　）则转移。

A. 结果溢出　　B. 结果为 0　　C. 结果为奇　　D. 结果为负

7. 当两个无符号数进行比较时，执行“JA　L”指令表示若（　　）则转移。

A. CF＝0 且 ZF＝0　　B. CF＝0 且 ZF＝1
C. CF＝1 且 ZF＝0　　D. CF＝1 且 ZF＝1

8. 当执行“CMP　AX，BX”指令进行比较后，执行“JG　L”指令转移到 L，则表明（　　）。

A. AX＞BX　　B. AX＝BX　　C. AX＜BX　　D. AX≥BX

9. 如果 0100H 字节单元存放条件转移指令的操作码，0101H 字节单元存放条件转移指令的相对位移量 13H，那么转移后的指令的偏移地址为（　　）。

A. 0102H　　B. 0113H　　C. 0115H　　D. 00FEH

10. 下列叙述中不正确的是（　　）。

A. 有符号数比较后的条件转移指令可采用 JG、JL、JGE 和 JLE 指令
B. 无符号数比较后的条件转移指令可采用 JA、JB、JNE 和 JBE 指令
C. 有符号数比较后判断溢出用 JO 和 JNO 指令
D. 无符号数比较后判断溢出用 JS 和 JNS 指令

单项选择题参考答案

1. A	2. B	3. D	4. A	5. D
6. A	7. A	8. A	9. C	10. D

5.4.2　多项选择题

1. 分支程序结构的形式有（　　）。

A. 单分支结构　　B. 双分支结构
C. 多分支结构　　D. 无分支结构

2. 在具有分支结构的程序中，其流程图一般有（　　）。

A. 判断转移框　　B. 循环框
C. 产生条件框　　D. 分支程序段框

3. 在下列的选项中，可以用来产生条件的选项有（　　）。

A. 数据传送指令　　B. 算术运算指令
C. 逻辑运算指令　　D. 转移指令

4. 在多分支程序设计中，常用的方法有（　　）。

A. 地址表法　　B. 计数法　　C. 转移表法　　D. 逻辑分解法

5. 执行完指令“ CMP　AX，100 ”后，如果要实现若 AX 寄存器中的无符号数高于 100 则转移到 L 的功能，应该执行的指令有（　　）。

A. JNS　L　　B. JNC　L　　C. JA　L　　D. JNBE　L

6. 执行完指令“ CMP　AX，0 ”后，如果要实现若 AX 寄存器中的内容为正数则转移到 L 的功能，应该使用的指令有（　　）。

A. JAE　L　　B. JGE　L　　C. JNS　L　　D. JNC　L

7. 如果用“ JS　P ”指令实现若 AX 寄存器中的内容为负数则转移的功能，产生条件的指令可用（　　）。

A. AND　AX，8000H　　B. CMP　AX，8000H
C. TEST　AX，8000H　　D. CMP　AX，0

8. 如果要实现若 AX 寄存器中的内容为 100H 则转移到 P 的功能，下列选项中可实现此功能的有（　　）。

A. CMP　AX，100H　　B. SUB　AX，100H
　　JZ　P　　　　JE　P
C. AND　AX，100H　　D. XOR　AX，100H
　　JZ　P　　　　JZ　P

9. 设 AX 寄存器中的内容为有符号数，如果要实现若 AX 寄存器中的内容小于 100H 则转移到 P 的功能，下列选项中可实现此功能的有（　　）。

A. CMP　AX，100H　　B. SUB　AX，100H
　　JL　P　　　　JNG　P
C. CMP　AX，100H　　D. SUN　AX，100H
　　JNGE　P　　　　JNAE　P

10. 如果在 TAB 字单元中存放的是某程序的入口地址，可以实现转移到该程序的入口地址的选项有（　　）。

A. LEA　BX，TAB　　B. MOV　BX，0
　　JMP　BX　　　　JMP　TAB[BX]
C. JMP　TAB　　D. JMP　DWORD PTR[TAB]

多项选择题参考答案

1. ABC	2. ACD	3. BC	4. ACD	5. CD
6. BC	7. ACD	8. ABCD	9. AC	10. AB

5.4.3 填空题

1. 当两个有符号数相减后，若要判断被减数大于等于减数则跳转，需用_____、_____或_____指令。

答：JGE　JNB　JNC

2. 在分支程序的结构中，其流程图一般由_____、_____、定向和_____四个部分组成。

答：产生条件　测试　标号

3. 转移表法分支程序设计中，其程序转移表中存放的是_____。

答：无条件转移指令

4. 地址表法分支程序设计中，其地址表中存放的是_____。

答：转移指令的地址

5. 逻辑分解法程序设计一般是利用多条_____指令实现多分支程序设计的。

答：条件转移

6. 条件转移指令是分支程序设计中最常用的指令之一，这类指令大体上可以分为三种，它们是_____条件转移，_____条件转移和_____条件转移指令。

答：简单　有符号数　无符号数

7. 条件转移指令的执行，是当条件_____时则转移。

答：满足

8. 在执行条件转移指令前，必须要执行产生条件码的指令，然后才能进行条件判断转移，而一般的数据传送指令_____影响条件码（标志位）。

答：不会（不）

9. 当两个数进行比较后，执行_____指令表示若结果为零则转移到 L。

答：JZ　L

10. 为了实现程序的多路分支，通常使用_____、_____和_____三种分支程序设计方法。

答：地址表法　转移表法　逻辑分解法

5.4.4　程序分析题

1. **现有程序如下：**

```
DATA    SEGMENT
X       DB      0ABH
FLAG    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        CMP     AL，0
        JGE     L
        MOV     FLAG，0
        JMP     EXIT
L：      MOV     FLAG，1
EXIT：   MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序执行完后** FLAG **中的内容是多少？**

【解答】（1）**判断** X，**当** X≥0 **时将** FLAG **置** 1，**否则清** 0。

（2）FLAG **中的内容为** 0。

2. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AH，1
        INT     21H
        CMP     AL，61H
        JB      N
        CMP     AL，7AH
        JA      N
        SUB     AL，20H
N：      MOV     BUF，AL
```

```
            MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**如果要将“**CMP　AL，7AH**”改为“**CMP　AL，7BH**”，则相应的指令“**JA　N**”应改为什么指令？**

【解答】（1）**将键盘输入的小写字母（**a~z**）转换为大写字母存放在** BUF **中，其他字符则不转换，也存放在** BUF **中。**

（2）JAE　L。

3. **现有程序如下：**

```
DATA        SEGMENT
A           DB      23
B           DB      0F0H
C           DB      0
DATA        ENDS
CODE        SEGMENT
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
            MOV     AL，A
            CMP     AL，B
            JZ      L
            JG      M
            MOV     C，－1
            JMP     EXIT
L：         MOV     C，0
            JMP     EXIT
M：         MOV     C，1
EXIT：      MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行完后，**C **中的内容是什么？**

【解答】（1）**计算表达式**

$$C=\begin{cases}-1 & A<B\\ 0 & A=B\\ 1 & A>B\end{cases}$$

（2）C 中的内容为 1。

4. **现有程序如下：**

```
DATA     SEGMENT
X        DW      7000H
Y        DW      9000H
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AX，X
         SUB     AX，Y
         JO      L
         MOV     DL，‘N’
         JMP     EXIT
L：      MOV     DL，‘Y’
EXIT：   MOV     AH，2
         INT     21H
         MOV     AH，4CH
         INT     21H
CODE     ENDS
         END     START
```

请回答：（1）该程序完成什么功能？

（2）程序执行完后，输出的结果是什么？

【解答】（1）判断 X－Y 是否产生溢出，有溢出则显示输出 Y，否则显示输出 N。

（2）显示输出 Y。

5. **现有程序如下：**

```
DATA     SEGMENT
BUF1     DB      ‘ABCDAC’
C1       =       $－BUF1
BUF2     DB      ‘C’
BUF3     DW      0
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     ES，AX
         LEA     DI，BUF1
         MOV     CX，C1
         MOV     AL，BUF2
```

```
            CLD
            REPNE   SCASB
            JNZ     N
            MOV     DL,‘Y’
            MOV     AH，2
            INT     21H
            DEC     DI
            MOV     BUF3，DI
            JMP     EXIT
N：         MOV     DL,‘N’
            MOV     AH，2
            INT     21H
EXIT：      MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行完后，**BUF3 **中的内容是什么？输出的结果是什么？**

【解答】（1）**搜索** BUF1 **中是否存放有** BUF2 **中的字符，有则显示输出** Y**，并将该字符在** BUF1 **中的存放位置送** BUF3 **保存；否则显示输出** N**。**

（2）BUF3 **中的内容为** 2**，显示输出** Y**。**

6. **现有程序如下：**

```
DATA        SEGMENT
BUF         DB      23，125，96
DATA        ENDS
CODE        SEGMENT
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
            LEA     SI，BUF
            MOV     AL，[SI]
            MOV     BL，[SI+1]
            MOV     CL，[SI+2]
            CMP     AL，BL
            JAE     N1
            XCHG    AL，BL
N1：        CMP     AL，CL
            JAE     N2
            XCHG    AL，CL
N2：        CMP     BL，CL
            JAE     N3
            XCHG    BL，CL
```

```
N3：    MOV     [SI]，AL
        MOV     [SI+1]，BL
        MOV     [SI+2]，CL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序运行后，BUF 中的内容依次是什么？

【解答】(1) 将 BUF 缓冲区中的三个数按从大到小的顺序排列。

(2) 125，96，23。

7. **现有程序如下：**

```
DATA    SEGMENT
X       DB      4FH
BUF     DB      2 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        MOV     CL，4
        SHR     AL，CL
        CMP     AL，9
        JBE     L
        ADD     AL，7
L：     ADD     AL，30H
        MOV     BUF，AL
        MOV     AL，X
        AND     AL，0FH
        CMP     AL，10
        JB      M
        ADD     AL，7
M：     ADD     AL，30H
        MOV     BUF+1，AL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序运行后，BUF 中的内容依次是什么？

【解答】（1）将 X 中的 2 位十六进制数转换成 ASCII 码存放在 BUF 中。

（2）34H，46H。

8. 现有程序如下：

```
DATA    SEGMENT
A       DW      1234H
B       DW      5678H
S       DB      'NO  SWAP!$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA
START:  MOV     AX,DATA
        MOV     DS,AX
        MOV     AH,1
        INT     21H
        CMP     AL,'Y'
        JNZ     M
        CMP     AL,'y'
        JNZ     M
        MOV     AX,A
        XCHG    AX,B
        MOV     A,AX
        JMP     N
M:      MOV     AH,9
        LEA     DX,S
        INT     21H
N:      MOV     AH,4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）若键盘输入“Y”时，A、B 中的值各为多少？

【解答】（1）当键盘输入字符‘Y’或‘y’时，则交换变量 A 和变量 B 中的内容，否则显示输出‘NO SWAP!’

（2）（A）＝5678H，（B）＝1234H。

9. 现有程序如下：

```
DATA    SEGMENT
X       DB      23H
Y       DB      45H
Z       DW      0
DATA    ENDS
```

```
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        CMP     AL，0
        JGE     L
        ADD     AL，Y
        MOV     AH，0
        ADC     AH，0
        MOV     Z，AX
        JMP     EXIT
L：      SUB     AL，Y
        MOV     AH，0
        SBB     AH，0
EXIT：   MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序计算的表达式是什么？**

（2）**程序运行后，**Z **中的内容是什么？**

【解答】（1）**计算的表达式是**

$$Z=\begin{cases}X-Y & X\geq 0\\ X+Y & X<0\end{cases}$$

（2）（Z）=0DEH **或**（Z）=-22H

10. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      0F3H
S       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，BUF
        TEST    AL，80H
        JZ      L
        NEG     AL
L：      MOV     S，AL
        MOV     AH，4CH
```

```
        INT     21H
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序运行后，S 中的内容是什么？

【解答】(1) 求 BUF 中的绝对值送 S。

(2) S 中的内容为 0DH。

11. **现有程序如下：**

```
DATA    SEGMENT
STR     DB      'ABCDEFGMABC'
C       =       $-STR
X       =       5
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA
START:  MOV     AX,DATA
        MOV     DS,AX
        MOV     ES,AX
        CLD
        LEA     SI,STR
        ADD     SI,X
        MOV     DI,SI
        INC     SI
        MOV     CX,C
        SUB     CX,X
        DEC     CX
        CMP     CX,0
        JBE     EXIT
        REP     MOVSB
EXIT:   MOV     AH,4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序运行完后，STR 中的内容是什么？

【解答】(1) 删除字符串 STR 中的第 6 个字符。

(2) 'ABCDEGMABC'

12. **现有程序如下：**

```
DATA    SEGMENT
```

```
X       DB      96H
Y       DB      0A2H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        CMP     AL，Y
        JG      L
        XCHG    AL，Y
L：      MOV     BL，AL
        MOV     CL，4
        SHR     AL，CL
        CMP     AL，9
        JBE     M
        ADD     AL，7
M：      ADD     AL，30H
        MOV     DL，AL
        MOV     AH，2
        INT     21H
        AND     BL，0FH
        CMP     BL，9
        JBE     N
        ADD     BL，7
N：      ADD     BL，30H
        MOV     AH，2
        MOV     DL，BL
        INT     21H
        MOV     DL，‘H’
        MOV     AH，2
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序运行后显示的内容是什么？

【解答】（1）比较 X 和 Y 中的有符号数，将其中的大者转换成十六进制数并输出。

（2）96H。

13. **现有程序如下：**

```
DATA    SEGMENT
```

```
D0      DB      ‘DATA  0$’
D1      DB      ‘DATA  1$’
D2      DB      ‘DATA  2$’
D3      DB      ‘DATA  3$’
X       DB      02H
TAB     DW      DIS0, DIS1, DIS2, DIS3
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START：  MOV     AX, DATA
        MOV     DS, AX
        LEA     BX, TAB
        MOV     AL, X
        AND     AL, 3
        MOV     AH, 0
        SHL     AX, 1
        ADD     BX, AX
        JMP     WORD PTR[BX]
DIS0：   LEA     DX, D0
        JMP     EXIT
DIS1：   LEA     DX, D1
        JMP     EXIT
DIS2：   LEA     DX, D2
        JMP     EXIT
DIS3：   LEA     DX, D3
EXIT:   MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行后显示输出的是什么？**

【解答】（1）**根据** X **中的值**（0、1、2、3）**分别显示不同字符串** DATA 0、DATA 1、DATA 2 **和** DATA 3。

（2）**显示输出** DATA 2。

14. **现有程序如下：**

```
DATA    SEGMENT
X       DB      3
Y       DB      45H
W       DB      0
DATA    ENDS
```

```
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     BX，OFFSET  TAB
        MOV     AL，X
        MOV     AH，0
        AND     AL，3
        SHL     AX，1
        ADD     BX，AX
        JMP     BX
EXIT：   MOV     AH，4CH
        INT     21H
TAB：    JMP     SHORT  M0
        JMP     SHORT  M1
        JMP     SHORT  M2
        JMP     SHORT  M3
M0：     MOV     AL，Y
        ADD     AL，8
        MOV     W，AL
        JMP     EXIT
M1：     MOV     AL，Y
        ADD     AL，16
        MOV     W，AL
        JMP     EXIT
M2：     MOV     AL，Y
        MOV     W，AL
        JMP     EXIT
M3：     MOV     W，100
        JMP     EXIT
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序运行后，**W **中的内容是什么？**

【解答】（1）**计算的表达式为**

$$W=\begin{cases} Y+8 & X=0 \\ Y+16 & X=1 \\ Y & X=2 \\ 100 & X\geq 3 \end{cases}$$

（2）W **中的内容为** 100。

15. **现有程序如下：**

```
DATA    SEGMENT
W       DB      ?
X       DB      25
Y       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        MOV     BL，W
        CMP     BL，0
        JZ      L1
        CMP     BL，1
        JZ      L2
        CMP     BL，2
        JZ      L3
        MOV     BL，AL
        MUL     AL
        JMP     EXIT
L1：     ADD     AL，100
L：      MOV     AH，0
        ADC     AH，0
        JMP     EXIT
L2：     ADD     AL，200
        JMP     L
L3：     SUB     AL，100
        MOV     AH，0
        SBB     AH，0
EXIT：   MOV     Y，AX
        INT     21H
CODE    ENDS
        END     START
```

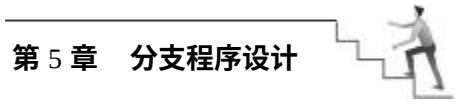
请回答：（1）**该程序完成什么功能？**

（2）**若** W **中的内容为** 5，**程序运行完后** Y **中的内容是多少？**

【解答】（1）**计算的表达式为**

$$Y=\begin{cases} X+100 & W=0 \\ X+200 & W=1 \\ X-100 & W=2 \\ X*X & W\geq 3 \end{cases}$$

（2）625。



5.4.5　程序填空题

1. **下列程序是判断** BUF **字缓冲区中有符号数的大小，当其数值大于**-100 **时，就将** FLAG **字节单元清** 0**，否则就将其置** 1**。请在程序的空格处填上适当的指令。**

```
        MOV     AX，BUF
        (1)
        (2)
        MOV     FLAG，0
        (3)
L：     MOV     FLAG，1
M：     MOV     AH，4CH
        INT     21H
```

【解答】(1) CMP　AX，-100
(2) JLE　L（**或** JNG　L）
(3) JMP　M

2. **下列程序是判断** X **和** Y **中有符号字数据的大小，当** X **中的数据大于** Y **中的数据时，就将** Z **置** 1**；当** X **中的数据小于** Y **中的数据时就将** Z **置**-1**；否则将** Z **清** 0**。请在程序的空格处填上适当的指令。**

```
        MOV     AX，X
        (1)
        (2)
        JG      M
        MOV     Z，－1
        JMP     EXIT
L：     MOV     Z，0
        JMP     EXIT
M：     (3)
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】(1) CMP　AX，Y
(2) JZ　L
(3) MOV　Z，1

3. **下列程序是判断两个无符号字数据** X、Y **的大小，当** X＞Y **时执行** X-Y**；当** X＜Y **时执行** Y-X**；当** X＝Y **时执行** X+Y**。其运算后的结果存放在** W **中。请在程序的空格处填上适当的指令。**

```
        MOV     AX，X
        MOV     BX，Y
        CMP     AX，BX
        (1)
```

```
        JB      M
        ADD     AX，BX
        ____(2)____
L：     ____(3)____
        JMP     EXIT
M：     XCHG    AX，BX
        SUB     AX，BX
EXIT：  MOV     W，AX
        MOV     AH，4CH
        INT     21H
```

【解答】（1）JA　　L
（2）JMP　　EXIT
（3）SUB　　AX，BX

4. **设 X、Y 是双精度的 32 位的无符号数变量，下列程序是完成 2 个数大小的比较，当 X 大于 Y 时，将字节变量 F 置 1，否则将字节变量 F 清 0。请在程序的空格处填上适当的指令。**

```
        MOV     AX，X＋2
        MOV     BX，X
        CMP     AX，Y＋2
        JB      L1
        ____(1)____
        CMP     BX，Y
        ____(2)____
L1：    ____(3)____
        JMP     EXIT
L2：    MOV     F，1
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】（1）JA　　L2
（2）JA　　L2
（3）MOV　　F，0

5. **下列程序是将 BUF 字节缓冲区中的 2 位十六进制数转换成 2 位十六进制数的 ASCII 码，并将转换结果存放在 Y 单元中。请在程序的空格处填上的适当的指令。**

```
        MOV     AL，BUF
        ____(1)____
        SHR     AL，CL
        ____(2)____
        JBE     L
        ADD     AL，7
L：     ADD     AL，30H
```

```
        MOV     Y，AL
        MOV     AL，BUF
              (3)
        CMP     AL，10
              (4)
        ADD     AL，7
M：     ADD     AL，30H
        MOV     Y+1，AL
        MOV     AH，4CH
        INT     21H
```

【解答】(1) MOV　　CL，4

(2) CMP　　AL，9

(3) AND　　AL，0FH

(4) JB　　M

6. **下列程序是判断键盘输入字符，若输入字符为数字，则将其数字的** ASCII **码送** BUF **保存；若为字母，则将其对应的大写字母送** BUF **保存；否则将** 0 **送** BUF **保存。请在程序的空格处填上适当的指令。**

```
        MOV     AH，1
        INT     21H
        CMP     AL，30H
              (1)
        CMP     AL，39H
              (2)
        CMP     AL，41H
        JB      M
        CMP     AL，5BH
              (3)
        CMP     AL，61H
        JB      M
        CMP     AL，7AH
              (4)
        SUB     AL，20H
L：     MOV     BUF，AL
        JMP     EXIT
M：     MOV     AL，0
        MOV     BUF，AL
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】(1) JB　　M

(2) JBE　　L

(3) JB　　L

(4) JA　　M

7. **下列程序是判断两个有符号字数据** X、Y，**当两个数都为正数时，则将** 2 **送往** W；**当两个数都为负数时，则将**-2 **送往** W；**当两个数异号时，则将** 0 **送往** W。**请在程序的空格处填上适当的指令。**

```
        MOV     AX，X
        MOV     BX，Y
        TEST    AX，8000H
              (1)
        TEST    BX，8000H
        JZ      L
P：     MOV     W，0
        JMP     EXIT
L：     MOV     W，2
        JMP     EXIT
M：     TEST    BX，8000H
              (2)
              (3)
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】（1）JNZ　　M
（2）JZ　　P
（3）MOV　　W，-2

8. **下列程序是比较字符串** STR1 **和** STR2 **中第一个不相等的字符，若查找到，则将** FLAG **字节单元置** 1，**并将查找的字符串** STR1 **的偏移地址送** ADR **字单元，否则** FLAG **清** 0。**请在程序的空格处填上适当的指令。**

```
        LEA     SI，STR1
              (1)
        MOV     CX，100
        CLD
              (2)
              (3)
        MOV     FLAG，0
        JMP     EXIT
L：     MOV     FLAG，1
              (4)
        MOV     ADR，SI
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】（1）LEA　　DI，STR2（**或** MOV　　DI，OFFSET STR2）
（2）REPE　　CMPSB（**或** REPZ　　CMPSB）
（3）JNZ　　L
（4）DEC　　SI

9. **下列程序是根据键盘输入的不同字符进行相应的处理。当输入为‘Y’或‘y’时，则将 BUF1 中 100 个字节的内容传送给 BUF2；当输入‘N’或‘n’时，则将 BUF2 中 100 个字节的内容传送给 BUF1；否则直接退出。请在程序的空格处填上适当的指令。**

```
        MOV     AH，1
        INT     21H
        CMP     AL，‘Y’
        JZ      L
              (1)
        JZ      L
        CMP     AL，‘N’
        JNZ     M
        CMP     AL，‘n’
        JNZ     M
        JMP     EXIT
L：     LEA     SI，BUF1
              (2)
        MOV     CX，100
        CLD
        REP     MOVSB
        JMP     EXIT
M：     LEA     SI，BUF2＋99
        LEA     DI，BUF1＋99
        MOV     CX，100
              (3)
        REP     MOVSB
EXIT：  MOV     AH，4CH
        INT     21H
```

【解答】（1）CMP　　AL，‘y’

（2）LEA　　DI，BUF2（**或** MOV　　DI，BUF2）

（3）STD

10. **下列程序是 X 和 Y 中的两个压缩型 BCD 码相加，如果相加结果超时 100，则显示 OVER 字符串中的内容；否则，将相加结果送 W，并显示 P 字符串中的内容。请在程序的空格处填上适当的指令。**

```
        MOV     AL，X
        ADD     AL，Y
              (1)
              (2)
        MOV     W，AL
        LEA     DX，P
        JMP     EXIT
L：     LEA     DX，OVER
```

```
EXIT：  MOV     AH，9
          （3）
        MOV     AH，4CH
        INT     21H
```

【解答】（1）DAA

（2）JC　　L

（3）INT　　21H

5.4.6 程序设计题

1. **编写一个程序，从键盘输入一个字符，若输入的字符为** Y（**或** y），**则将字变量** A **和** B **中的** 16 **位无符号数相乘，其结果存放在** F **字单元中；否则将执行** A/B **的运算，其结果的商存放在** F **中，余数存放在** F+2 **中。**

【解答】

```
DATA    SEGMENT
A       DW      ?
B       DW      ?
F       DW      2 DUP（0）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AH，1
        INT     21H
        CMP     AL，‘Y’
        JZ      L
        CMP     AL，‘y’
        JZ      L
        MOV     DX，0
        MOV     AX，A
        DIV     B
        MOV     F，AX
        MOV     F＋2，DX
        JMP     EXIT
L：     MOV     AX，A
        MUL     B
        MOV     F，AX
        MOV     F＋2，DX
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

2. **编写一个程序，判断 X 字节变量中的三个数，根据比较结果 FLAG 字节变量置 1：**

（1）若三个数都为 0，则 FLAG 置 3。
（2）若两个数为 0，则 FLAG 置 2。
（3）若一个数为 0，则 FLAG 置 1。
（4）若全不为 0，则 FLAG 置 0。

【解答】

```
DATA    SEGMENT
X       DB      35，95，0
FLAG    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，0
        CMP     X，0
        JNZ     L1
        INC     AL
L1：     CMP     X+1，0
        JNZ     L2
        INC     AL
L2：     CMP     X+2，0
        JNZ     L3
        INC     AL
L3：     MOV     FLAG，AL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

3. **设 X 为有符号数的字变量，编写一段程序，若 X 的绝对值大于 10，则将其绝对值送 Y 保存；否则将 X+10 的值送 Y 保存。**

【解答】

```
DATA    SEGMENT
X       DW      ？
Y       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，X
        TEST    AX，8000H
        JNS     L1
```

```
        NEG     AX
L1：    CMP     AX，10
        JBE     L2
        MOV     Y，AX
        JMP     EXIT
L2：    MOV     AX，X
        ADD     AX，10
        MOV     Y，AX
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

4. **编写一程序，判断** X、Y **字节变量中的数据，根据判断结果置** FLAG **字节变量的值。**

（1**）若两个数都为奇数，则将** FLAG **置** 2**。**

（2**）若两个数都为偶数，则将** FLAG **置** 0**。**

（3**）若两个数为一奇一偶，则将** FLAG **置** 1**。**

【解答】

```
DATA    SEGMENT
X       DB      ?
Y       DB      ?
FLAG    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，X
        MOV     BL，Y
        TEST    AL，1
        JZ      L1
        TEST    BL，1
        JZ      L2
        MOV     FLAG，2
        JMP     EXIT
L1：    TEST    BL，1
        JZ      L3
L2：    MOV     FLAG，1
        JMP     EXIT
L3：    MOV     FLAG，0
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

5. **编写一程序，统计 X 字节变量中为 1 的位数，并将统计的结果显示出来。**

【解答】

```
DATA    SEGMENT
X       DB      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
STAAT：  MOV     AX，DATD
        MOV     DS，AX
        MOV     DL，0
        MOV     AL，X
        MOV     CX，8
L1：     SHL     AL，1
        JNC     L2
        INC     DL
L2：     DEC     CX
        JNZ     L1
        ADD     DL，30H
        MOV     AH，2
        INT     21H
CODE    ENDS
        END     START
```

6. **编写一程序，将字符串缓冲区 BUF 中的小写字母转换成大写字母。**

【解答】

```
DATA    SEGMENT
BUF     DB      ‘AbAFAEdGma’
C       =       $－BUF
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE  DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，C
        LEA     BX，BUF
L：      MOV     AL，[BX]
        CMP     AL，61H
        JB      L1
        CMP     AL，7AH
        JA      L1
        SUB     AL，20H
        MOV     [BX]，AL
L1：     INC     BX
        DEC     CX
        JNZ     L
```

```
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

7. **编写一程序，将 BUF 缓冲区中的 2 位十六进制数转换成二进制数并输出。**

【解答】

```
DATA    SEGMETNT
BUF     DW      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，16
        MOV     BX，BUF
L：     SHL     BX，1
        JNC     L1
        MOV     DL，31H
        JMP     L2
L1：    MOV     DL，30H
L2：    MOV     AH，2
        INT     21H
        DEC     CX
        JNZ     L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

8. **编写一程序，将 X 变量中的 16 位二进制数转换成十六进制数并输出。**

【解答】

```
DATA    SEGMENT
X       DW      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     DH，4
        MOV     BX，X
L：     MOV     CL，4
        ROL     BX，CL
        MOV     DL，0F
        CMP     DL，9
        JBE     L1
```

```
        ADD     DL，7
L1：    ADD     DL，30H
        MOV     AH，2
        INT     21H
        DEC     DH
        JNZ     L
        MOV     AH，21H
        INT     21H
CODE    ENDS
        END     START
```

9. **用地址表法编写一程序，根据键盘输入的数值分别输出对应字符串：**

（1）输入“0**”时，显示输出“**INPUT　0**!”**

（2）输入“1**”时，显示输出“**INPUT　1**!”**

（3）输入“2**”时，显示输出“**INPUT　2**!”**

（4）输入其他数值时，显示“INPUT　ERROR**！**”

【解答】

```
DATA    SEGMENT
TAB     DW      D0，D1，D2，D3
DIS0    DB      ‘INPUT  0！$’
DIS1    DB      ‘INPUT  1！$’
DIS2    DB      ‘INPUT  2！$’
DIS3    DB      ‘INPUT  ERROR！$’
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AH，1
        INT     21H
        CMP     AL，30H
        JB      L
        CMP     AL，33H
        JB      L1
L：     MOV     AL，3
        JMP     L2
L1：    AND     AL，0FH
L2：    LEA     BX，TAB
        SHL     AL，1
        MOV     AH，0
        ADD     BX，AX
        JMP     WOR  PTR[BX]
D0：    LEA     DX，DIS0
        JMP     EXIT
```

```
D1：    LEA     DX，DIS1
        JMP     EXIT
D2：    LEA     DX，DIS2
        JMP     EXIT
D3：    LEA     DX，DIS3
EXIT：  MOV     AH，9
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

10. **编写一程序，将** BUF **字节缓冲区中的** 100 **字节按相反方向存放到** BUF1 **中。**

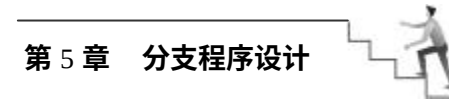
【解答】

```
DATA    SEGMENT
BUF     DB      100 DUP（?）
BUF1    DB      100 DUP（?）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，100
        LEA     SI，BUF
        LEA     DI，BUF1＋99
L：      MOV     AL，[SI]
        MOV     [DI]，AL
        DEC     DI
        INC     SI
        DEC     CX
        JNZ     L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

11. **编写一程序，用串比较指令，比较** STR1 **和** STR2 **字符串，若两个字符串完全相等，则输出‘** OK **！’；若不相等，则输出‘** ERROR **！’及** STR1 **中第一个不相等的字符。**

【解答】

```
DATA    SEGMENT
STR1    DB      ‘ABCDACde’
C1      ＝      $－STR1
STR2    DB      ‘ABCDEAcd’
C2      ＝      $－STR2
S1      DB      ‘OK！$’
S2      DB      ‘ERROR！$’
```



```
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     ES，AX
        CLD
        MOV     AX，C1
        CMP     AX，C2
        JNZ     DISE
        LEA     SI，STR1
        LEA     DI，STR2
        MOV     CX，C1
        REPZ    CMPSB
        JNZ     DISE
        LEA     DX，S1
        MOV     AH，9
        INT     21H
        JMP     EXIT
DISE：  LEA     DX，S2
        MOV     AH，9
        INT     21H
        MOV     DL，[SI－1]
        MOV     AH，2
        INT     21H
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

12. **编写一程序，比较两个有符号的字变量** X **和** Y**。**

（1**）若两个数都大于等于** 100**，则将两个数分别加** 1**。**

（2**）若两个数都不大于等于** 100**，则将两个数分别减** 1**。**

（3**）若两个数一个大于等于** 100**，另一个数不大于等于** 100**，则将两个数互换。**

【解答】

```
DATA    SEGMENT
X       DW      ?
Y       DW      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        CMP     X，100
```

```
        JL      L
        CMP     Y，100
        JL      M
        INC     X
        INC     Y
        JMP     EXIT
L：     CMP     Y，100
        JGE     M
        DEC     X
        DEC     Y
        JMP     EXIT
M：     MOV     AH，X
        XCHG    AX，Y
        MOV     X，AX
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

13. **假定 A、B、C 字节变量中存放的是三个相等的数据，但由于某种原因使其中的一个数据发生变化，试编写一程序找出三个数中变化的代码存放在 D 中，并将变化的代码单元恢复成原来的数据。**

【解答】

```
DATA    SEGMENT
A       DB      ?
B       DB      ?
C       DB      ?
D       DB      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        MOV     BL，B
        MOV     CL，C
        CMP     AL，BL
        JZ      L
        CMP     AL，CL
        JZ      M
        MOV     D，AL
        MOV     A，BL
        JMP     EXIT
L：     MOV     D，CL
        MOV     C，AL
```

```
        JMP     EXIT
M：     MOV     D，BL
        MOV     B，AL
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

14. **编写一程序，将 A 和 B 中的两位压缩型 BCD 码相加，并将相加的结果用十进制数输出。**

【解答】

```
DATA    SEGMENT
A       DB      45H
B       DB      67H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AL，A
        MOV     AH，0
        ADD     AL，B
        DAA
        ADC     AH，0
        MOV     BX，AX
        MOV     CL，4
        ROL     BX，CL
        MOV     CH，3
L：     MOV     CL，4
        ROL     BX，CL
        MOV     DL，BL
        AND     DL，0FH
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        DEC     CH
        JNZ     L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

第 6 章　循环程序设计

6.1 学习目的和要求

循环程序也是一种基本的、常用的和重要的程序结构形式之一，它是一种能重复处理同一类操作的程序设计方法。循环程序不但可以使程序的结构简单、清晰，而且还可以节省大量的内存空间。

通过本章的学习，要求能熟悉循环程序的基本结构形式及各组成部分的内容和功能；熟练地掌握单重循环和多重循环程序设计的方法和技巧；灵活地运用四种控制循环方法。

【学习重点】

1. 单重循环程序设计。
2. 多重循环程序设计。
3. 控制循环的方法。

【学习难点】

多重循环程序设计。

6.2 知识体系

6.2.1 知识体系结构

在本章中，所讲述的知识体系结构如图 6.1 所示。

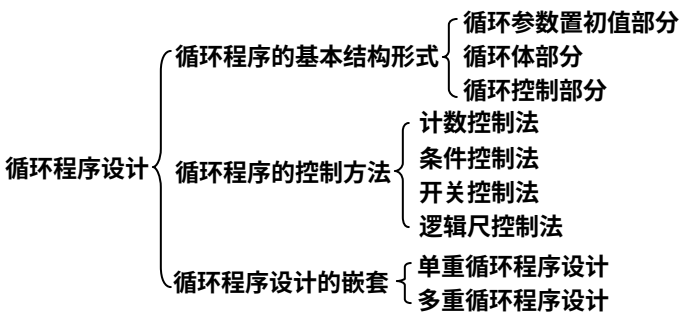


图 6.1　循环程序设计的知识体系结构

6.2.2　知识点与考核要求

1. 循环程序的概念和结构，应达到“理解”的层次。

（1）循环程序的含义和采用循环结构的必要性。
（2）循环程序的结构组成和各部分的作用。
（3）循环程序的两种基本结构形式，比较两种结构形式对控制循环的要求。

2. 单重循环程序设计，应达到“综合应用”的层次。

（1）单重循环程序的结构形式。
（2）单重循环程序设计的方法和技巧。重点在如何选择循环控制条件，如何选择工作参数和工作单元并设置初值，如何判别循环结束等。
（3）能够分析单重循环程序，能够应用单重循环程序设计方法编写程序。

3. 多重循环程序设计，应达到“简单应用”的层次。

（1）多重循环程序的结构形式，内层循环与外层循环遵守的层次结构规则，参数修改对各层的相互影响。
（2）多重循环程序的设计方法和技巧。重点在如何选择各层循环的操作参数和工作单元，如何考虑参数在层次间的相互影响，如何合理地选择内外层的控制循环结束条件，如何设置初值，如何判别内外层循环的结束。
（3）能够分析多重循环程序，能够应用多重循环编写程序。

4. 控制循环的方法，应达到“综合应用”的层次。

多种控制循环方法的控制原理，循环控制参数的选择和初值的设置，修改循环控制参数和判断循环结束条件的方法和技巧。

6.3　例题分析

例 1　现有程序如下：

```
DATA   SEGMENT
A      DB      12H，94H，56H，88H，01H，95H，32H，65H，47H，02H
B      EQU     $-A
C      DB      0
D      DB      0
DATA   ENDS
CODE   SEGMENT
       ASSUME  CS：CODE，DS：DATA
START：MOV     AX，DATA
       MOV     DS，AX
       MOV     DX，0
```

```
        LEA     BX，A
        MOV     CX，B
L：     MOV     AL，[BX]
        CMP     AL，0
        JGE     P
        INC     DL
        JMP     NEXT
P：     INC     DH
NEXT：  INC     BX
        LOOP    L
        MOV     C，DH
        MOV     D，DL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成的功能是什么？**

（2）**该程序执行完后** C、D **变量中的内容各是什么？**

解：根据该程序，可以画出如图 6.2 **所示的程序流程图。**

从图中可以看出，这是一个循环程序的结构。它首先置循环的初值，即将循环操作的首地址送 BX，**循环计数的次数送** CX，**将统计** A **数组中正数的个数** DH **和负数的个数** DL **寄存器的内容清** 0；**然后将** A **数组中的数依次一个一个取出来进行判断，若该数大于等于** 0 **则** DH **寄存器的内容加** 1，**否则** DL **寄存器的内容加** 1。**由于采用的是有符号数的判断，只要该数的最高位为** 1，**则该数为负数（即小于** 0），DL **的内容加** 1；**否则该数大于等于** 0，DH **的内容加** 1；**为了统计** A **数组中所有的数据，还需要对** BX **寄存器的内容进行修改，使指针指向下一个地址单元；最后判断循环是否结束，若循环还未结束则继续循环；否则退出循环，将统计的结果送对应的结果单元。所以该程序的功能是统计** A **数组中大于等于** 0 **的数的个数送** C **变量，小于** 0 **的数的个数送** D **变量。**

由于 A **数组中的元素共有** 10 **个，其中有** 7 **个元素的最高位为** 0，3 **个元素的最高位为** 1，**所以程序运行完后，**C **变量中的内容为** 7 **表示有** 7 **个大于等于** 0 **的数，**D **变量中的内容为** 3 **表示有** 3 **个负数。**

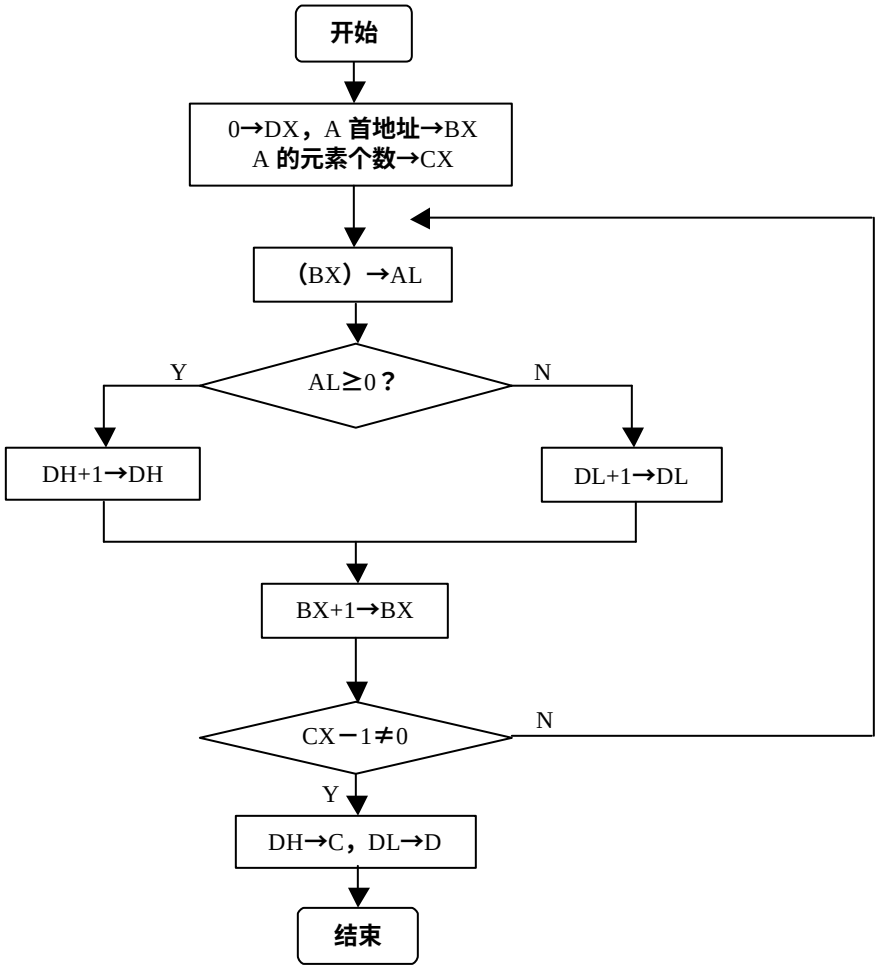


图 6.2　例 1 的程序流程图

例 2　现有程序如下：

```
DATA        SEGMENT
            ORG     $+20H
BUF         DB      10  DUP ('ABCDEFGHIJKLMNOPQRST')
BUF1        DW      BUF+10
STR         DB      20 DUP (0)
NUM         EQU     5
X           DB      100
DATA        ENDS
CODE        SEGMENT
            ASSUME  CS:CODE, DS:DATA
START:      MOV     AX, DATA
            MOV     DS, AX
            MOV     SI, BUF1
            LEA     DI, STR
```

```
            MOV     CX，NUM
            CMP     X，0
            JGE     L2
            ADD     SI，CX
            ADD     DI，CX
            DEC     DI
L1：        MOV     AL，[SI]
            MOV     [DI]，AL
            DEC     SI
            DEC     DI
            LOOP    L1
            JMP     RETDOS
L2：        MOV     AL，[SI]
            MOV     [DI]，AL
            INC     SI
            INC     DI
            LOOP    L2
RETDOS：    MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

请回答：（1）**该程序完成的功能是什么？**

（2）**该程序执行完后共传送了多少个数据？源操作数的首地址是什么？传送的数据内容是什么？**

（3）**若**（X）=98H，**该程序执行完后共传送了多少个数据？源操作数的首地址是什么？传送的数据内容是什么？**

解：在数据段中，由于数据段的开始中有一条“ $+20H **”伪指令，它将空出** 20H **个存储单元，即** BUF **的首地址是** 20H**。数据段中“** BUF1　DW　BUF+10 **”是一条定义地址的伪指令，**BUF1 **中的内容是** BUF **的首地址加** 10**，所以** BUF1 **中的内容是指向** BUF **缓冲区中存放字符** K **的地址。因此，字符** K **的地址为** 2AH**。**

在代码段中，以 BUF1 **中的内容作为源操作数的地址送** SI **寄存器，以** STR **的偏移地址作为目的操作数的地址，以** 5 **作为循环的次数，再根据** X **中的内容决定执行哪一段循环程序。其程序流程图如图** 6.3 **所示。**

在程序的执行过程中，由于 X **中的内容为** 100**，其值大于** 0**，所以程序转向** L2 **去执行。在此循环程序段中，它将源操作数** BUF **缓冲区中从** K **字符开始的连续** 5 **个字符传送到** STR **目的字符串中。因此，可以得到如下的解答。**

（1）**程序的功能为：将源操作数** BUF **缓冲区中从** K **字符开始的连续** 5 **个字符传送到** STR **目的字符串中。**

（2）**在程序的执行过程中，共传送了** 5 **个字符；源操作数的首地址为** DS：002AH**；传送的数据为‘** KLMNO **’（即** 4BH，4CH，4DH，4EH，4FH**）。**

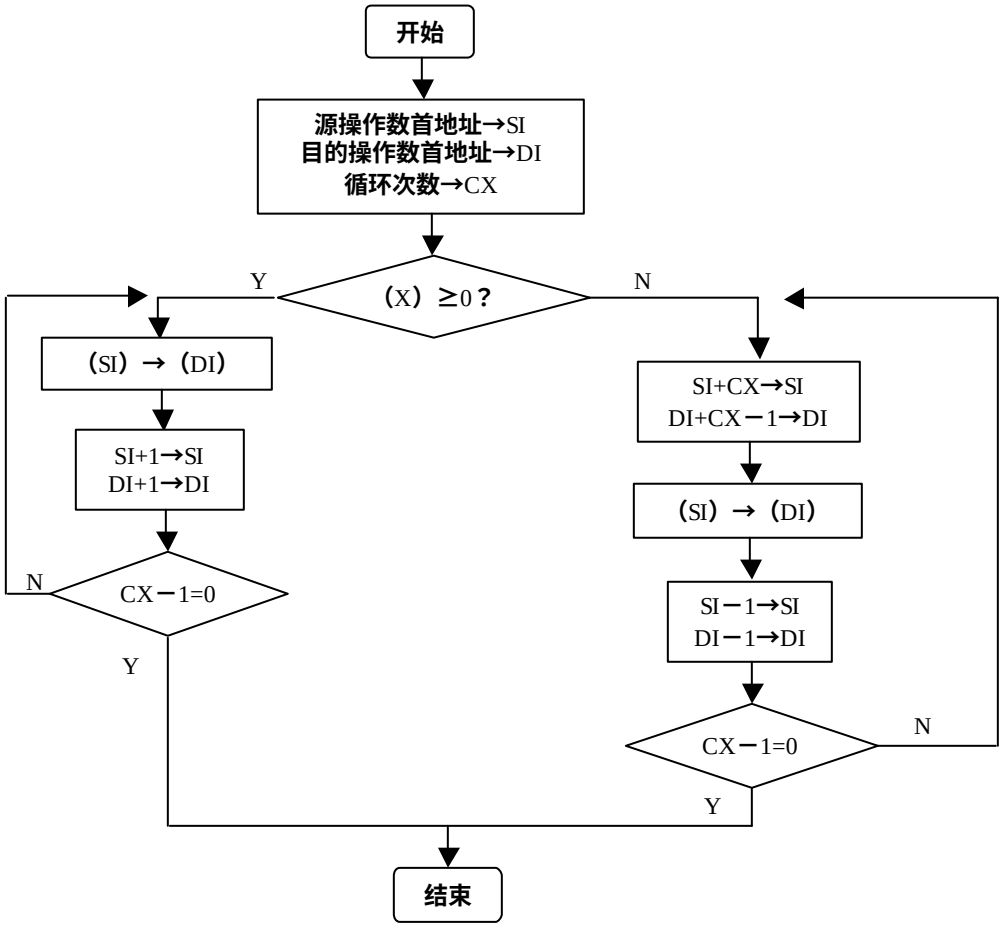


图 6.3　例 2 的程序流程图

（3）若（X）=98H，则它为负数，经判断后程序顺序往下执行。它将 SI 的内容加 5，使 SI 指向字符 P，DI 的内容加 4 指向 STR 的第 5 个单元。由于程序的循环次数是 5，源操作数的首地址为 DS：002FH，每传送一个字符则 SI、DI 的内容减 1，所以传送的字符为‘PONML’（即 4CH，4DH，4EH，4FH，50H）。

例 3　设 ARY 字数组中存放着 N 个有符号的字数据，下列程序是利用串指令取数，然后比较数据的大小，找出数组中的最大数和最小数并分别送 MX 和 NIN 的程序。请在下列程序的空格处填写适当的指令。

```
DATA      SEGMENT
ARY       DW      100  DUP （?）
N         =        （$-ARY）/2
MX        DW      0
NIN       DW      0
DATA      ENDS
CODE      SEGMENT
```

```
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
                      (1)
            LEA     SI，ARY
            MOV     CX，N
            MOV     AX，[SI]
            MOV     NIN，AX
            MOV     MX，AX
L1：                  (2)
            CMP     AX，MX
                      (3)
            MOV     MX，AX
                      (4)
L2：        CMP     AX，NIN
            JGE     L3
            MOV     NIN，AX
L3：        LOOP    L1
            MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

解：根据题目的要求和已给出的程序，可以画出程序的流程图如图 6.4 所示。

在程序填空中，一定要根据题目的具体要求，分析需要填写指令。利用程序的前后指令使用的寄存器、变量、数据的类型和使用的算法，合理地填写指令。

在本题中由于题目要求需要使用串操作指令，而在串操作指令中对于操作数的存取需要根据方向位 DF 来决定 SI、DI 的递增或递减。在程序中已经指明了存取数据的首地址是数组的首地址。如果要依次存取数据，必须从第一个数据开始依次存取。所以，在程序的空格（1）处应填写指令“ CLD ”，从第一个元素开始依次存取数据。

为了能够判断每一个数是否是该数组中的最小数和最大数，首先需要取一个数作为判断的标准，而在该程序中是以数组中的第一个数作为它的最大数和最小数分别存放在 MX 和 NIN 中。然后需要依次取 ARY 数组中的每一个数分别与 MX 和 NIN 中的数进行比较。由于在第（2）空格的后续指令中将 AX 寄存器中的内容作为比较的数据，同时第（2）空格要完成将数组 ARY 中的数送 AX 和地址修改的功能，因此，在第（2）空格处应填写“ LODSW ”指令。该指令除了将 SI 所指示单元的内容送 AX 外，还自动将 SI 的内容加 2，使 SI 指向下一个数据的地址单元。

在取出数据进行比较时，如果 AX 中的内容大于 MX 中的最大数，则应将 AX 中的内容作为最大数送 MX；如果 AX 中的内容小于 MX 中的最小数，则应将 AX 中的内容作为最小数送 NIN 单元；否则继续进行循环。在 L2 中的几条指令是完成将 AX 中的内容作为最小数送 NIN 单元的功能，所以在第（3）空格处应填写“ JLE　L2 ”指令。

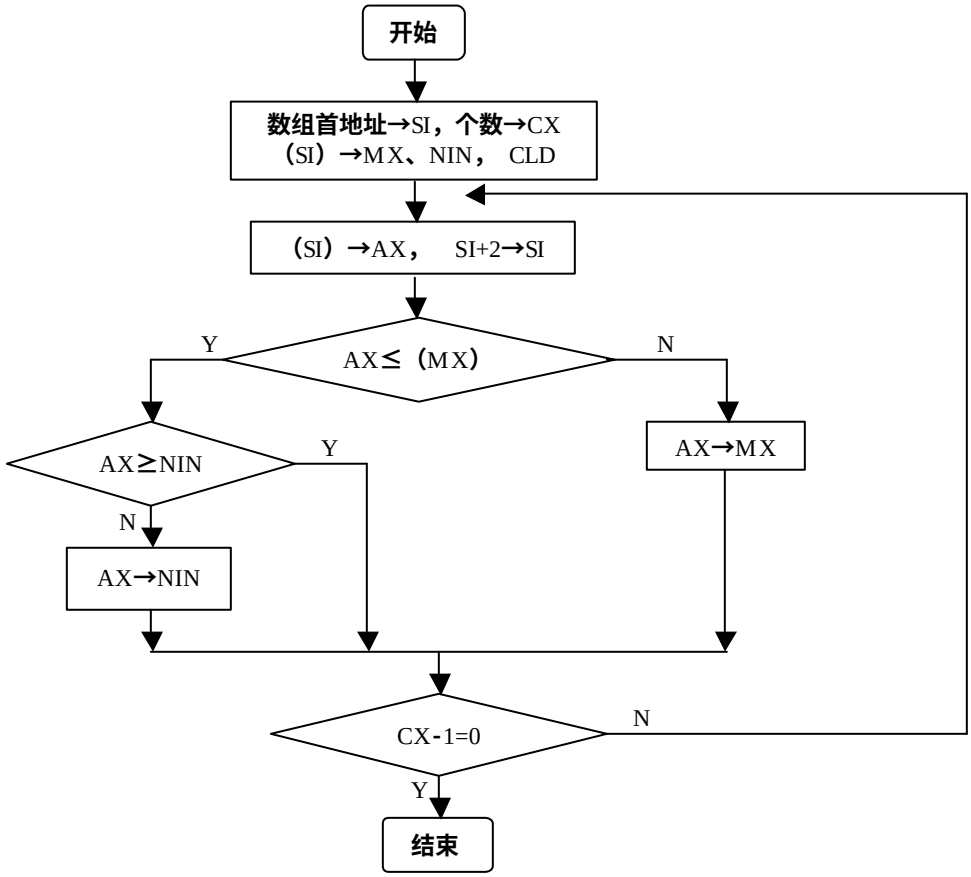


图 6.4　例 3 的程序流程图

当 AX 中的内容大于 MX 中的最大数时，它将 AX 中的内容作为最大数送 MX。为了使程序能循环执行，则应转移到 L3 继续循环。所以在第（4）空格处应填写“ JMP　L3 ”指令。

例 4　已知以 BUF 为首地址的字存储区中存放着 N 个有符号二进制数，试编写程序，将其中大于等于 0 的数依次送入以 BUF1 为首地址的字存储区中，小于 0 的数依次送入以 BUF2 为首地址的字存储区中。同时将大于等于 0 的数的个数送 A 字变量，将小于 0 的数的个数送 B 字变量。请在下列程序的空格处填写适当的指令。

```
DATA      SEGMENT
BUF       DW      23，123，-12，-210，45，0，90，-453
N         =        （$-BUF）/2
BUF1      DW      N  DUP（0）
BUF2      DW      N  DUP（0）
A         DW      0
B         DW      0
DATA      ENDS
CODE      SEGMENT
```

```
            ASSUME  CS：CODE，DS：DATA
START：     MOV     AX，DATA
            MOV     DS，AX
            LEA     BX，BUF
            LEA     SI，BUF1
            LEA     DI，BUF2
            MOV     A，0
            MOV     B，0
                    (1)
L0：        MOV     AX，[BX]
            CMP     AX，0
                    (2)
            MOV     [DI]，AX
                    (3)
            INC     B
            JMP     NEXT
L1：                (4)
            ADD     SI，2
            INC     A
NEXT：      ADD     BX，2
            LOOP    L0
            MOV     AH，4CH
            INT     21H
CODE        ENDS
            END     START
```

解：该程序的算法为：将 BUF 字存储区中的 N 个数依次取出，判断其值是否大于等于 0，若是，则送该数到 BUF1 存储区中，A 字变量的内容加 1；否则，送该数到 BUF2 存储区中，B 字变量的内容加 1。如此重复，直至 N 个数处理完毕。由此可见，实现以上算法的程序应是一个循环程序，其循环次数为 N。

根据程序需要实现的功能和以上的算法分析，可以画出程序流程图如图 6.5 所示。同时，寄存器的分配如下：

BX：**BUF 存储区的地址指针，初值指向 BUF。**

SI：**大于等于 0 的数存储区的地址指针，初值指向 BUF1。**

DI：**小于 0 的数存储区的地址指针，初值指向 BUF2。**

CX：**循环计数器，初值为 BUF 区中数据的个数 N。**

AX：**用来暂存待判断的数。**

通过程序流程图可以很明显地看出，在第（1）空格中应填写指令“MOV　CX，N”，将 N 送 CX 作为循环的次数。

当取出一个数后，如果该数大于等于 0 则将此数送 BUF1 缓冲区保存，使计数器 A 加 1，并将缓冲区指针地址加 2 指向下一个数据。因在 L1 循环语句指令中有计数器 A 加 1，缓冲区指针地址加 2 指向下一个数据的指令。所以在程序的第(2)空格处应填写“JGE　L1”指令。

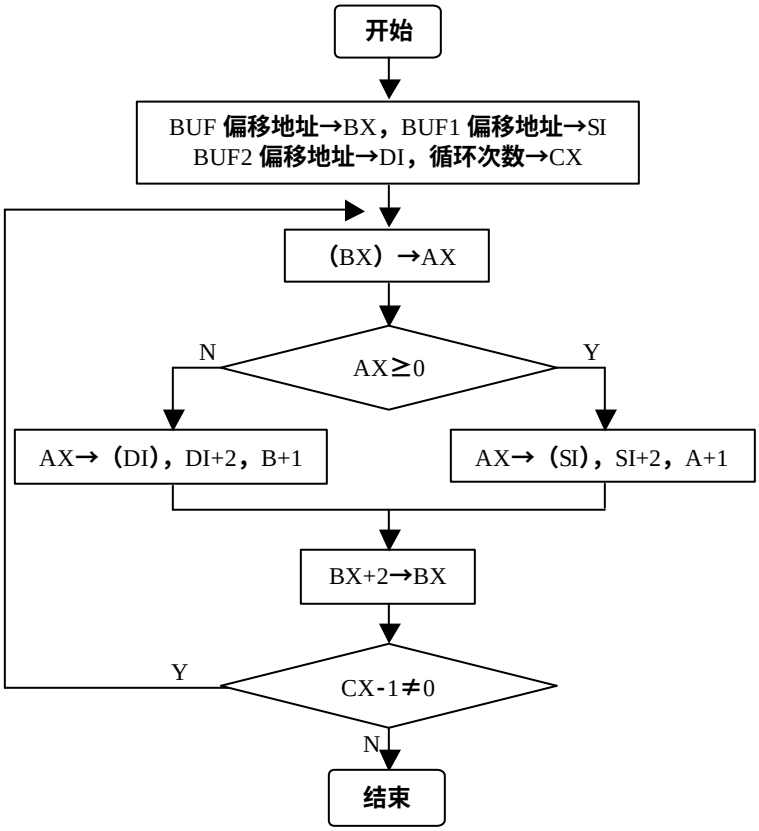


图 6.5　例 4 程序流程图

如果取出的数小于 0，应将此数送 BUF2 缓冲区，将计数器 B 的个数加 1，并将缓冲区指针地址加 2 指向下一个数据。所以在第（3）空格处应填写“ ADD　DI，2 ”指令。

由以上分析可知，在 L1 循环语句中的指令是完成大于等于 0 的数据的处理。即当某一数据为大于等于 0 时，应将此数送 BUF1 缓冲区，将计数器 A 的值加 1，并将缓冲区指针地址加 2 指向下一个数据。所以，第（4）空格应填写“ MOV　[SI]，AX ”指令。

例 5　设有一字数组 A，第一个字单元存放的是数组元素的个数，从第二个字单元开始存放的是数组的元素。下列程序是用 SCAS 串搜索指令搜索字数组 A 中与 X 字变量中所有相等的元素。若找到该元素则用 MOVS 指令从数组中删除该元素；否则，数组中的元素不变。请在下列程序的空格处填写适当的指令。

```
DATA      SEGMENT
A         DW      10，23，123，-12，-210，45，0，90，-453，9A5H，67BH
X         DW      0
DATA      ENDS
CODE      SEGMENT
          ASSUME  CS：CODE，DS：DATA
START：   MOV     AX，DATA
          MOV     DS，AX
                  (1)
```

```
          MOV     AX，X
          CLD
L：       MOV     CX，A
          LEA     DI，A+2
                  (2)
          JNZ     OK
          CMP     CX，0
          JZ      NO
          MOV     SI，DI
                  (3)
                  (4)        MOVSW
          DEC     A
          JMP     L
NO：              (5)
OK：      MOV     AH，4CH
          INT     21H
CODE      ENDS
          END     START
```

解：根据题目的具体要求，可以画出程序的流程图，如图 6.6 所示。

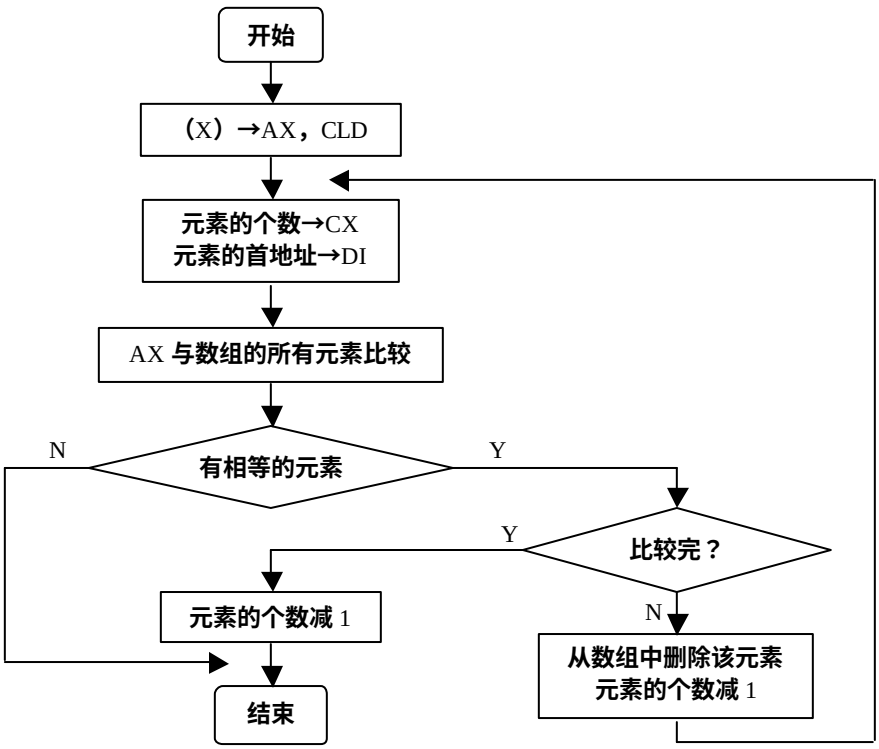


图 6.6　例 5 程序流程图

从题目的要求和程序的流程图可知，在数组 A 中有多少个与（X）相等的元素是一个未知数。因此该程序是一个循环次数未知的循环程序。

在使用串操作指令时，要特别注意其源操作数的地址在 DS：SI 中，目的操作数的地址在 ES：DI 中。在程序中已经将数据段的段地址送 DS，但没有送 ES，所以在第（1）空格处应填写指令“ MOV　ES，AX ”。

因为在串搜索前，已经将要搜索的值送 AX，方向位 DF 清 0，也将要搜索的个数送 CX，搜索的目的起始地址送 DI。又因为题目的要求是要查找数组 A 中与（X）相等的元素，当不相等时，应该继续查找。因此在使用串操作搜索指令时，应使用“ REPNZ/REPNE ”重复前缀。所以，在第（2）空格处应填写“ REPNZ　SCASW ”或“ REPNE　SCASW ”指令。

在退出“ SCASW ”指令时，有两个退出的条件，一是已经找到相等的元素，标志位 ZF 为 1；二是查找完所有的元素没有相等的元素，标志位 ZF 为 0。此时可以用标志位 ZF 来判断是否找到相等的元素，但不能依据 CX 寄存器中的内容是否为 0 来判断是否找到相等元素。这是因为在查找的过程中有可能最后一个元素就是相等的元素，而且 CX 也是为 0 的。当 ZF 位为 0 时，表示在数组中已经没有相等的元素，此时应退出循环。只有当 ZF 位为 1 时，才表示在数组中有相等的元素，此时，应将此元素从数组中删除。但在删除元素时有两种情况，第一种情况是找到的元素不是数组的最后一个元素，此时需要从数组中删除该元素。根据题目的要求，需要使用串操作指令“ MOVS ”完成功能。此时应该将源操作数的地址指针指向相等元素的下一个数，目的操作数地址指针指向相等元素，以 CX 中还剩余未查找完的个数作为数据传送的个数，再用串操作指令进行数据传送，达到删除元素的目的。因此，要实现 DI 指向相等元素，SI 指向相等元素的下一个数，在第（3）空格处应填写指令“ SUB　DI，2 ”，在第（4）空格处应填写“ REP ”。第二种情况是找到的元素是数组的最后一个元素，此时只须将数组元素的个数减 1 就达到了删除的目的。因此在第（5）空格处应填写“ DEC　A ”指令。

例 6　试编制一个程序，把字变量 X 中的 16 位二进制数以十六进制数的形式在屏幕上显示。

解：根据题意，我们应该把 X 中的 16 位二进制数从左到右分成 4 组，每 4 位为一组。用 4 位二进制数表示 1 位十六进制数，则应在屏幕上显示 4 位十六进制数。显然这可以利用循环结构来完成，每次循环显示 1 位十六进制数，共计循环 4 次，因而循环次数是已知的，循环计数值为 4。循环体中则应包括从二进制数到所显示字符的 ASCII 码的转换，以及字符的显示。后者可以使用 DOS 功能调用来实现。其程序流程图如图 6.7 所示。

这里采用了循环移位的方法把所要显示的 4 位二进制数移到最右面（最低 4 位），以便作数字到字符的转换工作。另外，由于数字 0～9 的 ASCII 码为 30H～39H，将数字转换为 ASCII 码需加 30H；而字母 A～F 的 ASCII 码为 41H～46H，将字母转换为 ASCII 码需加 37H。所以在把 4 位二进制数转换为 ASCII 码的时候，需要判断 4 位二进制数所表示的是 0～9 还是 A～F。首先判断低 4 位二进制数是字母还是数字，如果为数字，则只需将数字加 30H 就是 0～9 的 ASCII 码；如果为字符 A～F，则需要加 37H 才是 A～F 的 ASCII 码。只有将 4 位二进制数转换为 ASCII 码后才能显示出正确的十六进制数。

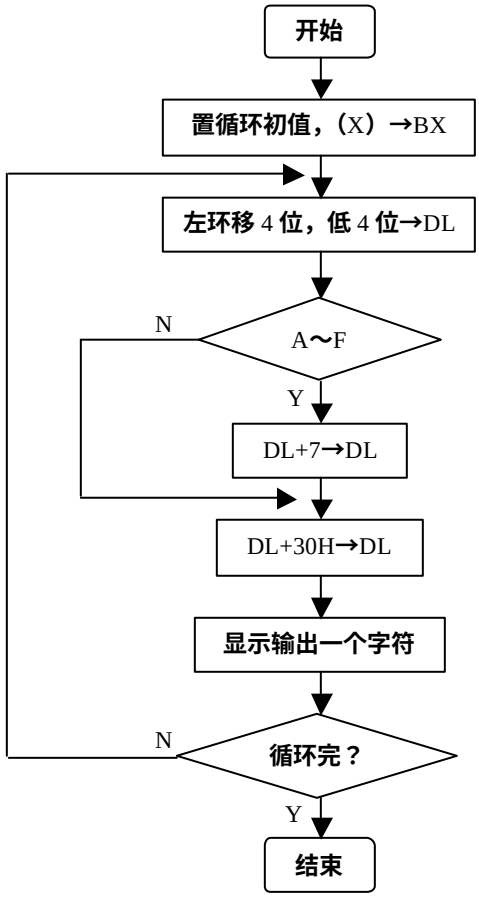


图 6.7　例 6 的程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
X       DW      4F59H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CH，4          ；置循环初值
        MOV     BX，X
L：      MOV     CL，4          ；BX 中的内容左环移 4 位
        ROL     BX，CL
        MOV     DL，BL         ；低 4 位送 DL
        AND     DL，0FH        ；DL 的低 4 位转换为 ASCII 码
        CMP     DL，10
        JB      NEXT
```

```
        ADD     DL，7
NEXT：  ADD     DL，30H
        MOV     AH，2          ；输出并显示
        INT     21H
        DEC     CH
        JNZ     L              ；修改循环次数，未完则转 L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 7　设 STR 字符串是以 0 结尾。试编写一个把字符串中的所有大写字母改为小写字母的程序，并将转换后的字符串显示输出。

解：根据题意分析，要完成其功能，需编写一个循环程序。由于字符串是以 0 结尾的，所以字符串的长度是一个未知数，它的循环次数也是不确定的，需根据字符串尾这个条件来控制程序的循环。如果是大写字母，需将对应字母的 ASCII 码加 20H；如果是其他字符，则其字符保持不变。编写的程序清单如下：

```
DATA    SEGMENT
STR     DB      ‘HOW  arE YoU!’，0        ；假设的字符串
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     SI，OFFSET STR          ；取字符串开始地址
AGAIN： MOV     DL，[SI]                ；取一字符
        OR      DL，DL                  ；是否到字符串尾？
        JZ      OK                      ；到字符串尾，转 OK
        CMP     DL，‘A’                 ；否则，判断是否为大写字母
        JB      NEXT                    ；否，转继续
        CMP     DL，‘Z’
        JA      NEXT                    ；否，转继续
        ADD     DL，20H                 ；是大写字母，则改为小写字母
        MOV     [SI]，DL                ；送回到字符串中
NEXT：  MOV     AH，2
        INT     21H
INC     SI                              ；调整指针
        JMP AGAIN                       ；继续
OK：    MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 8　设 A 变量中存放着 10 个双字长的二进制数，请用 386 及其后继机型的相应指令编写将 A 送 B 的程序。

解：在进行 32 位数据传送的程序设计过程中，首先应该使用 32 位的数据传送指令。当数据传送后，为了使地址指针指向下一个数据，需对地址指针进行修改。由于一个 32 位的数据占用 4 个字节，在作地址修改时，需将地址加 4。

```
        .MODEL  SMALL
        .386
        .STACK  200H
        .DATA
A       DD      10  DUP (?)
B       DD      10  DUP (?)
        .CODE
START:  MOV     AX,@DATA
        MOV     DS,AX
        MOV     CX,10           ;置循环初值
        LEA     ESI,A
        LEA     EDI,B
L:      MOV     EAX,[ESI]       ;传送一个双字长的二进制数
        MOV     [EDI],EAX
        ADD     ESI,4           ;地址修改
        ADD     EDI,4
        LOOP    L               ;循环控制
        MOV     AX,4C00H
        INT     21H
        END     START
```

例 9　设 A、B 变量中分别存放着 10 个双字长的多精度二进制数，其中低地址存放的是低位数，高地址存放的是高位数。请用 386 及其后继机型的相应指令编写将 A＋B 送 C 的程序。

解：由于 A、B 变量中分别存放着 10 个双字长的二进制数，在进行 32 位数据相加的程序设计过程中，首先应该使用 32 位的数据加法指令。当进行双字长的数据加法时，不但要对两个 32 位数相加，同时还要加低位的进位，即用带进位指令。当 32 位数据相加完毕，为了使地址指针指向下一个数据，需对地址进行修改。由于一个 32 位的数据占用 4 个字节，在作地址修改时，需将地址加 4。当 10 个双字长的二进制数相加完毕后，最后的进位需加到最高地址中。编写的程序清单如下：

```
        .MODEL      SMALL
        .386
        .STACK      200H
        .DATA
A       DD          10  DUP  (?)
B       DD          10  DUP  (?)
```

```
C       DD          11  DUP (0)
        .CODE
START:  MOV     AX, @DATA
        MOV     DS, AX
        CLC                     ;置循环初值
        MOV     CX, 10
        LEA     ESI, A
        LEA     EDI, B
        LEA     EBX, C
L:      MOV     EAX, [ESI]      ;两个双字长的二进制数相加
        ADC     EAX, [EDI]
        MOV     [EBX], EAX      ;结果送目的单元
        ADD     ESI, 4          ;地址修改
        ADD     EDI, 4
        ADD     EBX, 4
        LOOP    L               ;循环控制
        MOV     EAX, 0          ;最高进位的处理
        ADC     EAX, 0
        MOV     [EBX], EAX
        MOV     AX, 4C00H
        INT     21H
        END     START
```

例 10　设 ARY 中存放着以 0 作为结束的双字数组。请用 386 及其后继机型的相应指令编写出统计数组中大于 0 和小于 0 的元素个数，并将统计的个数分别存放在 P 和 N 变量中的程序。

解：根据题目的要求，需要用 386 及其后继机型的数据传送指令和比较指令等来完成所要求的功能。编写的程序清单如下：

```
        .MODEL      SMALL
        .386
        .STACK      200H
        .DATA
ARY     DD          100  DUP (?)
P       DW          0
N       DW          0
        .CODE
START:  MOV     AX, @DATA
        MOV     DS, AX
        MOV     CX, 100
        LEA     ESI, ARY
L:      MOV     EAX, [ESI]
        CMP     EAX, 0
        JZ      EXIT
```

```
        JNS     PLUS
        INC     N
        JMP     CONT
PLUS：  INC     P
CONT：  ADD     SI，4
        LOOP    L
EXIT：  MOV     AX，4C00H
        INT     21H
        END     START
```

例 11　在以 BUF 为首地址的字数组中存放着 N 个有符号数，试编制程序使该数组中的数按照从大到小的次序排序。

解：该程序是一个多重循环程序。实现数据排序的算法有多种，下面介绍三种数据排序算法的基本思想和相应的程序。

方法一：冒泡排序算法。从第一个数开始与第二个数进行比较，如果第一个数大于等于第二个数，则不做任何操作；如果第一个数小于第二个数，则使两个数交换位置。然后再将第二个数与第三个数进行比较，如果第二个数大于等于第三个数，则不做任何操作；如果第二个数小于第三个数，则使两个数交换位置。依次重复，直到 N 个数相互比较完为止。由此可以看出，第一遍的比较需要做 N-1 次，此时将一个最小数放到最后，而将较大的数往前移动。在做第二遍的比较时，由于最小的数已经放到了最后，所以第二遍比较只需要考虑 N-1 个数的比较，即只需要比较 N-2 次。依此类推，第三遍则只需要做 N-3 次比较……总共最多 N-1 遍比较就可以完成排序。图 6.8 表示了冒泡排序算法的程序流程图。

编写的程序清单如下：

```
DATA    SEGMENT
BUF     DW      23，23，-7，-56，14，48，72，80，1，44，20，10
N       =       （$-BUF）/2
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，N               ；置循环次数
        DEC     CX
L1：    MOV     DI，CX
        MOV     BX，0               ；置位移量初值
L2：    MOV     AX，BUF[BX]         ；比较大小关系
        CMP     AX，BUF[BX+2]
        JGE     NEXT                ；大于等于则不交换，转 NEXT
        XCHG    AX，BUF[BX+2]       ；小于则交换
        MOV     BUF[BX]，AX
NEXT：  ADD     BX，2               ；修改位移量
        DEC     DI                  ；一遍未比较完则转 L2
```

```
        JNZ     L2
        LOOP    L1                    ；N-1 遍未比较完则转 L1
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

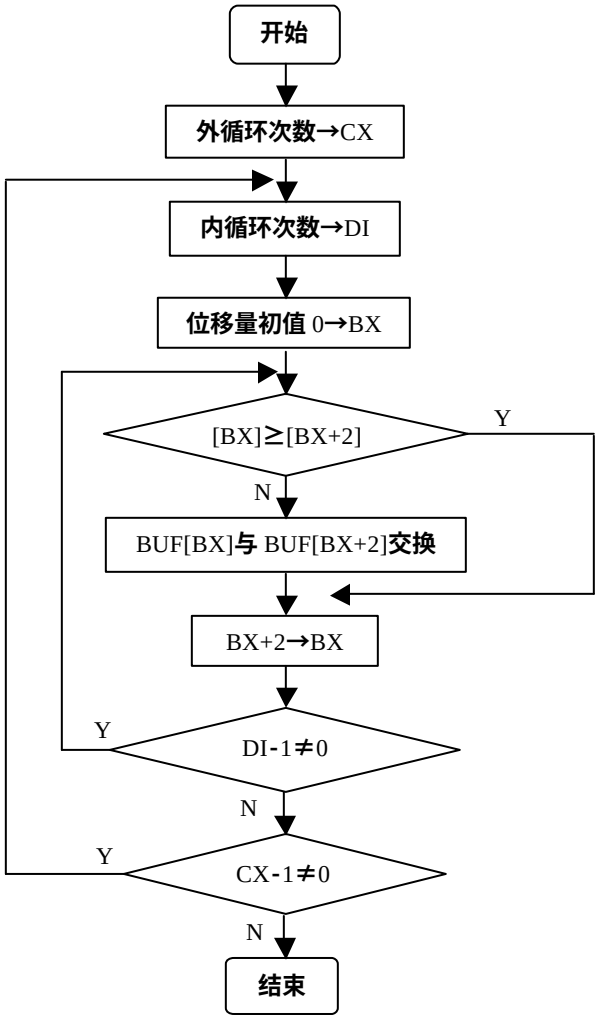


图 6.8　例 11 的冒泡排序算法的程序流程图

方法二：逐一比较法。其算法是将第一个存储单元中的数与其后 N-1 个存储单元中的数逐一比较。每次比较之后，总是把大者放第一个存储单元之中，经过 N-l 次比较之后，N 个数中最大者存入到第一个存储单元之中；接着将第二个存储单元中的数与其后的 N-2 个存储单元中的数逐一比较，每次比较之后，总是把大者放在第二个存储单元之中，经过 N-2 次比较之后，N 个数中的第二大者存入到第二个存储单元之中；如此重复下去，当最后两个存储单元之中的数比较完之后，从大到小排列的顺序就实现了。由此可以看出，第一遍的比较需做 N-1 次，此时将一个最大数放到第一个存储单元。第二遍比较只需要做

N-2 次，N 个数中的第二大者存入到第二个存储单元。依此类推，第三遍则只需要做 N-3 次比较……总共最多 N-1 遍比较就可以完成排序。

编写的程序清单如下：

```
DATA    SEGMENT
BUF     DW      23，23，-7，-56，14，48，72，80，1，44，20，10
N       =       （$-BUF）/2
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，N               ；置循环次数
        DEC     CX
        MOV     SI，0               ；置位移量初值
L1：     MOV     BX，CX
        MOV     DI，SI
        ADD     DI，2
        MOV     AX，BUF[SI]
L2：     CMP     AX，BUF[DI]         ；比较大小关系
        JGE     NEXT                ；大于等于则不交换，转 NEXT
        XCHG    AX，BUF[DI]         ；小于则交换
        MOV     BUF[SI]，AX
NEXT：   ADD     DI，2               ；修改位移量
        DEC     BX                  ；一遍未比较完则转 L2
        JNZ     L2
        ADD     SI，2               ；修改位移量
        DEC     CX                  ；N-1 遍未比较完则转 L1
        JNZ     L1
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

方法三：快速排序法。在上述的两种算法中，不管数组的原始排列情况如何，算法保证只要做 N-1 遍比较，总可以达到排序的目的。显然，在很多情况下，数组的比较遍数并未达到 N-1 遍就已经排序完毕，但程序必须继续运行到 N-l 遍才能结束。为了提高程序的运行速度，可以采用一种根据条件判断来决定外循环的方法。其方法是在程序运行时设立一个交换标志，每次进入内循环前就将交换标志清 0，如果在内循环中有交换操作就将该标志位置 1，此时表示数据未排好序，需进一步排序；如果在内循环中没有交换操作则该标志保持 0 不变，表示数据已排好序，不需要进一步排序。在每次内循环结束后，可以测试交换标志，如果该标志为 0 则再一次进入外循环进一步排序；如果该标志为 1，则说明上一遍比较未引起交换操作，数组已排序完毕，这样就可以立即结束外循环了。当然，这种方法在数组已排序完毕后会多做一遍比较。但在多数情况下，其比较遍数会少于 N-l 遍，

因而算法效率较高。这种算法的程序如下所示。

```
DATA    SEGMENT
BUF     DW      23，23，-7，-56，14，48，72，80，1，44，20，10
N       =       （$-BUF）/2
F       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，N           ；置循环次数
        DEC     CX
        MOV     F，1            ；交换标志位置 1
L1：    MOV     DI，CX
        CMP     F，1
        JNZ     EXIT            ；未交换则转 EXIT
        MOV     F，0            ；交换标志位清 0
        MOV     BX，0           ；置位移量初值
L2：    MOV     AX，BUF[BX]     ；比较大小关系
        CMP     AX，BUF[BX+2]
        JGE     NEXT            ；大于等于则不交换，转 NEXT
        XCHG    AX，BUF[BX+2]   ；小于则交换
        MOV     BUF[BX]，AX
        MOV     F，1            ；交换标志位置 1
NEXT：  ADD     BX，2           ；修改位移量
        DEC     DI              ；一遍未比较完则转 L2
        JNZ     L2
        LOOP    L1              ；N-1 遍未比较完则转 L1
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

例 12　已知 M×N 阶矩阵 A 的元素 Aij 按行序存放在以 A 为首地址的字节存储区中，试编写程序，求每行元素之和 S_i 并存放在 S 数组中。

解：每行元素之和 S_i 的计算公式为：

$$S_i = \sum_{i=1}^{n} A_{ij} \qquad (i=1，2，\cdots，m)$$

即

$S_1=A_{11}+A_{12}+\cdots+A_{1n}$

$S_2=A_{21}+A_{22}+\cdots+A_{2n}$

$$S_m=A_{m1}+A_{m2}+\cdots+A_{mn}$$

根据题意，可以画出程序流程图如图 6.9 所示。

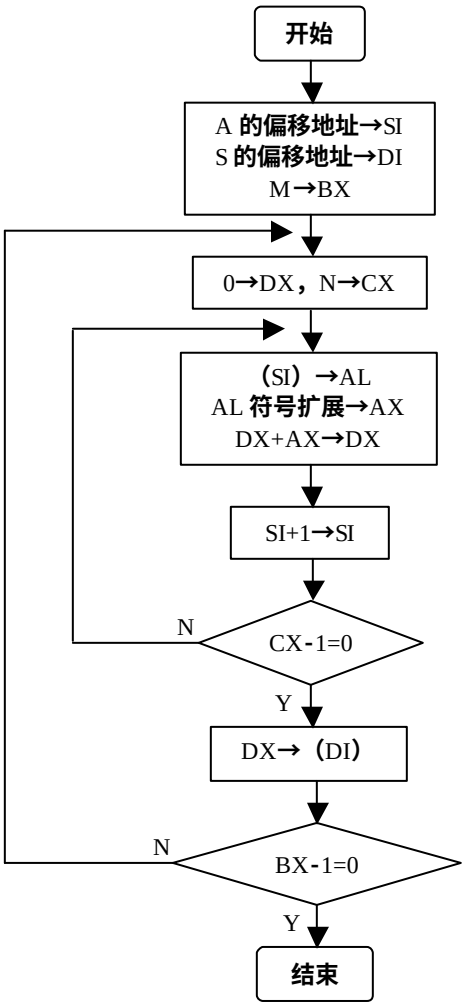


图 6.9　例 12 的算法程序流程图

编写的程序清单如下：

```
DATA    SEGMENT
A       DB      11, 12, 13, 14, 15, 16
        DB      17, 18, 19, 20, 21, 22
        DB      23, 24, 25, 26, 27, 28
        DB      29, 30, 31, 32, 33, 34
        DB      35, 36, 37, 38, 39, 40
M       =       5
N       =       6
```

```
S       DW      M  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，A
        LEA     DI，S
        MOV     BX，M
L1：     MOV     DX，0
        MOV     CX，N
L2：     MOV     AL，[SI]
        CBW
        ADD     DX，AX
        INC     SI
        LOOP    L2
        MOV     [DI]，DX
        ADD     DI，2
        DEC     BX
        JNZ     L1
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

6.4 练习题与参考答案

6.4.1 单项选择题

1. **循环指令中作为循环次数的寄存器是（　　）。**

   A. AX　　B. BX　　C. CX　　D. DX

2. **循环指令的转移范围是（　　）。**

   A. -128～127　　B. 0～255　　C. -32768～32767　　D. 0～65535

3. **循环指令“ LOOP ”的操作是（　　）。**

   A. CX **先减** 1，**然后再判断** CX，**若** CX=0 **则转，否则继续执行**
   B. CX **先减** 1，**然后再判断** CX，**若** CX≠0 **则转，否则继续执行**
   C. **先判断** CX，**若** CX≠0 **则** CX-1 **后转，否则继续执行**
   D. **先判断** CX，**若** CX≠0 **则转，否则** CX-1 **后继续执行**

4. 在循环程序设计过程中，将循环的次数送 CX 寄存器的操作是在哪一部分完成的？

A. 置初值部分　　B. 循环工作部分
C. 循环修改部分　　D. 循环控制部分

5. 在循环程序设计过程中，修改循环的次数是在哪一部分完成的？

A. 置初值部分　　B. 循环工作部分
C. 循环修改部分　　D. 循环控制部分

6. 在循环程序设计中，如果循环的次数事先无法确定，则应采用的循环控制方法是（　　）。

A. 条件控制法　　B. 计数控制法
C. 逻辑尺控制法　　D. 开关控制法

7. 下列描述正确的是（　　）。

A. 在多重循环程序中，内层循环只能有一个
B. 在汇编语言程序设计中，二重循环程序之间的关系可以交错
C. 在汇编语言程序设计中，每次循环应返回到置初值部分之前
D. 在汇编语言程序设计中，每次循环应返回到置初值部分之后

8. 如果在循环程序中的循环次数是 0～255 之间的一个数，则在程序设计时应采用的循环程序结构是（　　）。

A. 先执行后判断　　B. 先判断后执行
C. 判断和执行同时进行　　D. 判断和执行无先后顺序

9. 在多重循环程序中，每次通过外层循环进入内层循环时，其内层循环的初始条件（　　）。

A. 不必考虑　　B. 必须重新设置　　C. 必须置 1　　D. 须清 0

10. 在下列的程序段中，执行循环次数最多的是（　　）。

A.
```
    MOV   CX，1
    MOV   AX，0
L：INC   AX
    LOOP  L
```

B.
```
    MOV   CX，-1
    MOV   AX，0
L：INC   AX
    LOOP  L
```

C.
```
    MOV   CX，0
    MOV   AX，0
L：INC   AX
    LOOP  L
```

D.
```
    MOV   CX，0FFFFH
    MOV   AX，0
L：INC   AX
    LOOP  L
```

单项选择题参考答案

1. C	2. A	3. B	4. A	5. D
6. A	7. D	8. B	9. B	10.C

6.4.2　多项选择题

1. **循环程序结构的三个主要组成部分是（　　）。**

A. **置初值部分**　　B. **工作部分**　　C. **循环控制部分**　　D. **结束部分**

2. **在循环程序中，循环控制的方法有（　　）。**

A. **条件控制法**　　B. **计数控制法**
C. **逻辑尺控制法**　　D. **开关控制法**

3. **在循环程序设计中，如果循环的次数事先已确定，则采用的计数循环控制方法有（　　）。**

A. **正计数法**　　B. **倒计数法**　　C. **条件控制法**　　D. **开关控制法**

4. **循环指令“** LOOPNE　L **”控制循环结束的条件有（　　）。**

A. BX=0？　　B. CX=0？　　C. ZF=0？　　D. CF=0？

5. **在下列程序段中，程序运行后** AX **寄存器中的结果相同的是（　　）。**

A.
```
        MOV   CX，-1
        MOV   AX，0
   L：  INC   AX
        LOOP  L
```

B.
```
        MOV   CX，1
        MOV   AX，0
   L：  INC   AX
        LOOP  L
```

C.
```
        MOV   CX，2
        MOV   AX，0
   L：  INC   AX
        LOOP  L
```

D.
```
        MOV   CX，3
        MOV   AX，0
   L：  DEC   CX
        JZ    N
        INC   AX
        JMP   L
   N：  …
```

6. **在下列程序段中，程序运行循环次数相同的是（　　）。**

A.
```
        MOV   CX，10
   L：  …
        LOOP  L
```

B.
```
        MOV   CX，10
   L：  DEC   CX
        JZ    N
```

```
                                …
                                JMP     L
                                N：…
```

C.

```
        MOV     CX，10
L：     DEC     CX
        …
        CMP     CX，0
        JNZ     L
```

D.

```
        MOV     CX，10
L：     CMP     CX，0
        JZ      N
        …
        DEC     CX
        JMP     L
N：…
```

7. **在下列程序段中，若** 0≤X≤100，**程序运行后，循环次数完全相同的是（　　）。**

A.

```
        MOV     CX，X
L：     …
        LOOP    L
```

B.

```
        MOV     CX，0
L：     …
        INC     CX
        CMP     CX，X
        JNZ     L
```

C.

```
        MOV     CX，0
L：     INC     CX
        …
        CMP     CX，X
        JNZ     L
```

D.

```
        MOV     CX，X
L：     CMP     CX，0
        JZ      N
        …
        DEC     CX
        JMP     L
N：…
```

8. **在循环程序的工作部分，它一般包括（　　）。**

A. **置循环参数初值部分**　　B. **重复操作的程序段**
C. **循环参数的修改部分**　　D. **循环次数的控制部分**

9. **在多重循环程序中，内层循环（　　）。**

A. **只能是一个循环程序段**　　B. **可以是一个循环程序段**
C. **可以是两个循环程序段**　　D. **可以是多个循环程序段**

10. **在循环程序设计中，如果需要根据不同的条件执行不同的循环程序段，一般不采用的方法是（　　）。**

A. **条件控制法**　　B. **计数控制法**
C. **逻辑尺控制法**　　D. **开关控制法**

多项选择题参考答案

1. ABC	2. ABCD	3. AB	4. BC	5. CD
6. ACD	7. ABC	8. BC	9. BCD	10. ABC

6.4.3　填空题

1. 送循环的次数是在循环程序结构的_____ 完成的。
答：循环参数置初值部分（置初值部分）

2. 送循环操作的首地址是在循环程序结构的_____ 完成的。
答：循环参数置初值部分（置初值部分）

3. 修改循环的次数是在循环程序结构的_____ 完成的。
答：循环控制部分

4. 循环程序的基本结构主要由_____ 、_____ 和_____ 三个部分组成的。
答：循环参数置初值部分　循环工作部分　循环控制部分

5. 循环程序的结构有_____ 种。先执行循环体，后判断是否需要继续循环的循环结构称为_____ ；先判断是否需要执行循环体，后执行循环体的循环结构称为 _____ 。
答：2　先执行后判断　先判断后执行

6. 循环控制部分的连续两条指令“ DEC　CX ”和“ JNZ　L ”可以用一条_____指令来代替。
答：LOOP　L

7. 在循环当中套循环的程序称为 _____ 程序。
答：多重循环

8. 在循环程序的循环控制方法中，若循环次数未知时，循环结构通常采用的控制方法称为_____ 。若一个循环结构中包含有若干个循环体，每个循环体对应一个循环条件，这种循环结构通常采用的控制方法称为_____ 。
答：条件控制法　开关控制法

9. 所谓倒计数法就是先将计数器的初值设置成_____，每执行一次循环体后计数器就_____ ，然后判断循环是否应结束。
答：循环次数　减 1

10. 所谓正计数法就是先将计数器的初值设置成 0，每执行一次循环体后计数器就_____ ，然后与规定的循环次数比较，判断循环是否应结束。
答：加 1

11. 如果将计数器的初值设置成-N，每执行一次循环体后计数器就加 1，直到计数器的结果为_____ 时循环结束。

答：0

6.4.4 程序分析题

1. 现有程序如下：

```
DATA    SEGMENT
BUF     DB      ‘AI39*5867Jfe=KJYKGNGK339385’
C       =       $-BUF
N1      DB      0
N2      DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，C
        LEA     SI，BUF
L：      MOV     AL，[SI]
        CMP     AL，30H
        JB      NEXT
        CMP     AL，39H
        JA      NEXT
        INC     N1
        JMP     CONT
NEXT：   INC     N2
CONT：   INC     SI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）该程序执行完后，N1 和 N2 的内容各是多少？

【解答】（1）该程序完成的功能是：统计 BUF 缓冲区中数字字符的个数和非数字字符的个数并分别存放在 N1 和 N2 中。

（2）该程序执行完后，N1 和 N2 的内容分别是 12（0CH）和 15（0FH）。

2. 现有程序如下：

```
DATA    SEGMENT
BUF     DB   ‘ER39*5867JgeewFGHYUO9385’
```

```
C     =    $-BUF
DATA       ENDS
CODE       SEGMENT
           ASSUME  CS：CODE，DS：DATA
START：    MOV     AX，DATA
           MOV     DS，AX
           MOV     CX，C
           LEA     SI，BUF
L：        MOV     AL，[SI]
           CMP     AL，61H
           JB      NEXT
           CMP     AL，7AH
           JA      NEXT
           SUB     AL，20H
           MOV     [SI]，AL
NEXT：     INC     SI
           LOOP    L
           MOV     AH，4CH
           INT     21H
CODE       ENDS
           END     START
```

请回答：（1）该程序完成什么功能？

（2）该程序执行完后，BUF 缓冲区的内容是什么？

【解答】（1）该程序完成的功能是：将 BUF 缓冲区中的小写字母转换为大写字母。

（2）该程序执行完后，BUF 缓冲区中的内容是

‘ER39*5867JGEEWFGHYUO9385’。

3. **现有程序如下：**

```
DATA       SEGMENT
BUF        DB      47H，0A4H，93H，38，-23，55H，251，0，78H
C          =       $-BUF
BUF1       DB      C  DUP (0)
BUF2       DB      C  DUP (0)
DATA       ENDS
CODE       SEGMENT
           ASSUME  CS：CODE，DS：DATA
START：    MOV     AX，DATA
           MOV     DS，AX
           MOV     CX，C
           LEA     SI，BUF
           LEA     DI，BUF1
           LEA     BX，BUF2
L：        MOV     AL，[SI]
```

```
        CMP     AL，0
        JGE     P
        MOV     [BX]，AL
        INC     BX
        JMP     NEXT
P：     MOV     [DI]，AL
        INC     DI
NEXT：  INC     SI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**该程序执行完后，**BUF1 **和** BUF2 **缓冲区的内容各是什么？**

【解答】（1）**该程序完成的功能是：将** BUF **缓冲区的内容按大于等于** 0 **的数和小于** 0 **的数分别送** BUF1 **和** BUF2。

（2）BUF1 **缓冲区的内容为：**47H、38、55H、0、78H。

BUF2 **缓冲区的内容为：**0A4H、93H、-23、251。

4. **现有程序如下：**

```
DATA    SEGMENT
BUF1    DB      100 DUP（?）
BUF2    DB      200 DUP（0）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：MOV      AX，DATA
        MOV     DS，AX
        MOV     CX，100
        LEA     SI，BUF1
        LEA     DI，BUF2+100
L：     MOV     AL，[SI]
        MOV     [DI]，AL
        INC     SI
        INC     DI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）如果将指令“LEA　DI，BUF2+100”改为“LEA　DI，BUF2”之后，程序执行完后的结果如何？

【解答】（1）该程序完成的功能是：将 BUF1 缓冲区中的 100 个字节单元中的内容送 BUF2 缓冲区的后 100 个字节单元中。

（2）将 BUF1 缓冲区中的 100 个字节单元中的内容送 BUF2 缓冲区的前 100 个字节单元中。

5. 现有程序如下：

```
DATA     SEGMENT
BUF1     DB      100 DUP (?)
BUF2     DB      100 DUP (0)
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS:CODE, DS:DATA
START:   MOV     AX, DATA
         MOV     DS, AX
         MOV     CX, 100
         LEA     SI, BUF1
         LEA     DI, BUF2
L:       MOV     AL, [SI]
         CMP     AL, 100
         JB      NEXT
         MOV     [DI], AL
         INC     DI
NEXT:    INC     SI
         LOOP    L
         MOV     AH, 4CH
         INT     21H
CODE     ENDS
         END     START
```

请回答：（1）该程序完成什么功能？

（2）如果将指令“JB　NEXT”改为“JAE　NEXT”，则程序执行完后的结果如何？

【解答】（1）该程序完成的功能是：将 BUF1 缓冲区中大于等于 100 的数送 BUF2 缓冲区中。

（2）将 BUF1 缓冲区中小于 100 的数送 BUF2 缓冲区中。

6. 现有程序如下：

```
DATA     SEGMENT
BUF1     DB      100 DUP (?)
SUM      DW      0
```

```
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，100
        LEA     SI，BUF1
        MOV     AX，0
L：      ADD     AL，[SI]
        ADC     AH，0
        INC     SI
        LOOP    L
        MOV     SUM，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）如果删除指令“ADC　AH，0”，则程序执行结果将如何？

【解答】（1）该程序完成的功能是：将 BUF1 缓冲区的 100 个单元的内容累加，其结果送 SUM。

（2）如果在程序中没有“ADC　AH，0”指令，其结果的最高进位将会丢失。

7. **现有程序如下：**

```
DATA    SEGMENT
BUF1    DB      32H，64H，72H，38H，68H，9AH，89H，0
SUM     DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF1
        MOV     AX，0
L：      CMP     [SI]，BYTE PTR 0
        JZ      NDO
        ADD     AL，[SI]
        ADC     AH，0
        INC     SI
        JMP     L
NDO：    MOV     SUM，AX
        MOV     AH，4CH
        INT     21H
```

```
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序执行完后，共执行了多少次相加运算？

【解答】(1) 该程序完成的功能是：将 BUF1 缓冲区中的以 0 为结尾的若干个单元的内容累加，其结果送 SUM。

(2) 程序执行完后，共执行了 7 次相加运算。

8. **现有程序如下：**

```
DATA    SEGMENT
BUF1    DB      32H，54H，78H，73H，64H，29H，68H，03H
C       =       $-BUF
SUM     DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     BX，C
        LEA     SI，BUF1
L：      MOV     DL，[SI]
        MOV     CL，4
        SHR     DL，CL
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        MOV     DL，[SI]
        AND     DL，0FH
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        DEC     BX
        JNZ     L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：(1) 该程序完成什么功能？

(2) 程序执行完后，显示的结果如何？

【解答】(1) 该程序完成的功能是：将 BUF1 缓冲区中压缩的 BCD 码转换成十进制数并显示。

（2）在显示器上显示的结果是 3254787364296803。

9. 现有程序如下：

```
DATA    SEGMENT
BUF     DW      5A63H
SUM     DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CL，0
        MOV     AX，BUF
L：      AND     AX，AX
        JZ      EXIT
        SAL     AX，1
        JNC     L
        INC     CL
        JMP     L
EXIT：   MOV     SUM，CL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序执行完后，SUM 单元的内容是多少？

【解答】（1）该程序完成的功能是：统计 BUF 字单元中为 1 的位数送 SUM 单元。

（2）程序执行完后，SUM 单元的内容为 7。

10. 现有程序如下：

```
DATA    SEGMENT
A       DB      91，12，13，14，15，16
B       DB      37，18，19，20，21，22
C       DB      7  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，A
        LEA     DI，B
        LEA     BX，S
        MOV     CX，6
```

```
        CLC
L：     MOV     AL，[SI]
        ADC     AL，[DI]
        MOV     [BX]，AL
        INC     SI
        INC     DI
        INC     BX
        LOOP    L
        MOV     AL，0
        ADC     AL，0
        MOV     [BX]，AL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）如果删除指令“ADC　AL，0”，则程序执行完后的结果将如何？

【解答】（1）该程序完成的功能是：将数组 A 和数组 B 的内容相加，其结果送数组 C。

（2）如果在程序中没有“ADC　AL，0”指令，其结果的最高进位将会丢失。

11. **现有程序如下：**

```
STACK   SEGMENT STACK
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BIN     DW      7462
BUF     DB      6 DUP (0)，0DH，0AH，‘$’
TEN     DW      10
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        MOV     AX，BIN
        OR      AX，AX
        JNS     PLUS
        NEG     AX
        MOV     BUF，‘-’
        JMP     NEXT
PLUS：  MOV     BUF，‘+’
NEXT：  MOV     CX，5
```

```
L1：    MOV     DX，0
        DIV     TEN
        PUSH    DX
        LOOP    L1
        MOV     CX，5
        LEA     BX，BUF+1
L2：    POP     AX
        ADD     AL，30H
        MOV     [BX]，AL
        INC     BX
        LOOP    L2
        LEA     DX，BUF
        MOV     AH，9
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序执行完后，显示的结果如何？

【解答】（1）该程序完成的功能是：将 BIN **中的二进制数转换为十进制数显示输出。**

（2）程序执行完后，显示的结果为 7462**。**

12. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      23H，65H，28H，91H，66H
STR     DB      10 DUP (0)
DISBUF  DB      10 DUP (0)，0DH，0AH，' $ '
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
        LEA     DI，STR
        LEA     BX，DISBUF+9
        MOV     DX，5
L：      MOV     AL，[SI]
        AND     AL，0FH
        ADD     AL，30H
        MOV     [DI]，AL
        INC     DI
        MOV     [BX]，AL
```

```
        DEC     BX
        MOV     AL，[SI]
        MOV     CL，4
        SHR     AL，CL
        ADD     AL，30H
        MOV     [DI]，AL
        INC     DI
        MOV     [BX]，AL
        DEC     BX
        DEC     DX
        JNZ     L
        LEA     DX，DISBUF
        MOV     AH，  9
        INT     21H
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序执行完后，显示的结果如何？

【解答】（1）该程序完成的功能是：将 BUF **缓冲区中的压缩型** BCD **码转换为非压缩型** BCD **码，并** STR **缓冲区保存和送** DISBUF **缓冲区显示输出。**

（2）程序执行完后，显示的结果为 6691286523**。**

13. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      ‘46DDR6$54DR$$5YD6757$8F6￥8ITI0￥$’，0
COUNT   DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
        MOV     COUNT，0
L：     MOV     AL，[SI]
        CMP     AL，0
        JZ      EXIT
        CMP     AL，‘$’
        JNZ     NEXT
        INC     COUNT
NEXT：  INC     SI
        JMP     L
```

```
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序执行完后，**COUNT **的内容为何值？**

【解答】（1）**该程序完成的功能是：统计以** 0 **为结尾的** BUF **字符串缓冲区中字符** $ **出现的次数，并将统计的结果送** COUNT **单元保存。**

（2）**程序执行完后，**COUNT **结果为** 5。

14. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      100 DUP（?）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
        MOV     CX，100
        MOV     AX，0
        MOV     BX，1
L：     MOV     [SI]，AX
        MOV     DX，AX
        ADD     AX，BX
        MOV     BX，DX
        ADD     SI，2
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序执行完后，**BUF **中前** 10 **个数的内容为何值？**

【解答】（1）**该程序完成的功能是：在** BUF **缓冲区中生成斐波纳契数列值。**

（2）**程序执行完后，**BUF **中的内容为** 0，1，1，2，3，5，8，13，21，34。

15. **现有程序如下：**

```
DATA    SEGMENT
BUF     DB      100 DUP（?）
FLAG    DB      0
```

```
ADDR    DW      0
SUM     DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
        MOV     CX，100
        MOV     AX，0
        MOV     SI，0
L：      ADD     AX，BUF[SI]
        JO      OVER
        ADD     SI，2
        LOOP    L
        MOV     FLAG，0
        SUB     SI，2
        JMP     EXIT
OVER：   MOV     FLAG，1
EXIT：   MOV     SUM，AX
        SHR     SI，1
        INC     SI
        MOV     ADDR，SI
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）程序中若没有指令“ INC　SI **”，程序执行完后可能会出现什么结果？**

【解答】（1）该程序完成的功能是：将 BUF **缓冲区的内容进行累加，若在累加过程中出现溢出，则** FLAG **置** 1**，累加的单元个数送** ADDR**，累加的和数送** SUM**；否则** FLAG **清** 0**，累加的单元个数送** ADDR**，累加的和数送** SUM**。**

（2）程序中若没有指令“ INC　SI **”，程序执行完后累加的单元个数可能会少一个。**

16. **现有程序如下：**

```
RW      MACRO   A，B
        MOV     AH，A
        LEA     DX，B
        INT     21H
        ENDM
DATA    SEGMENT
```

```
BUF1    DB      100, 0, 100 DUP (0)
BUF2    DB      ‘PLEASE  INPUT:’
BUF3    DB      0AH, 0DH,‘$’
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        RW      9, BUF2
        RW      10, BUF1
        RW      9, BUF3
        MOV     CL, BUF1+1
        MOV     CH, 0
        MOV     SI, CX
        MOV     AH, 2
L:      MOV     DL, BUF1[SI+1]
        INT     21H
        DEC     SI
        LOOP    L
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）该程序完成什么功能？

（2）若输入的字符为‘12345’时，输出的结果是什么？

【解答】（1）该程序完成的功能是：将键盘输入的字符串按逆序输出。

（2）输出的结果为 54321。

17. **现有程序如下：**

```
DATA    SEGMENT
ARY     DW      100 DUP (?)
C       =       ($-ARY)/2
MX      DW      0
NIN     DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, C
        DEC     CX
        LEA     DI, ARY
        MOV     AX, [DI]
```

```
        MOV     BX, AX
L：     ADD     DI, 2
        CMP     [DI], BX
        JAE     L1
        MOV     BX, [DI]
        JMP     NEXT
L1：    CMP     [DI], AX
        JBE     NEXT
        MOV     AX, [DI]
NEXT：  LOOP    L
        MOV     MX, AX
        MOV     NIN , BX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

请回答：（1）**该程序完成什么功能？**

（2）**程序执行完后，**MX、NIN **中各存放何值？**

【解答】（1）**该程序完成的功能是：找出数组** ARY **中的最大值和最小值并分别送** MX、NIN。

（2）MX **中存放最大值，**NIN **中存放最小值。**

6.4.5　程序填空题

1. **下列程序是统计** STR **字符串中出现** X **变量中存放的字符的次数，并将统计的结果送** COUNT **单元。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
STR     DB      ‘CAGEYGUUIYO4637DYU7R656SR’
C       =       $－ARY
X       DB      ‘U’
COUNT   DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START：MOV      AX, DATA
        MOV     DS, AX
        MOV     COUNT, 0
        ____(1)____
        MOV     CX, C
        MOV     AL, X
L：     CMP     AL, [SI]
        ____(2)____
```

```
        INC     COUNT
NEXT：  INC     SI
        _____(3)_____
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）LEA　SI，STR　**或**　MOV　SI，OFFSET STR

（2）JNZ　NEXT

（3）LOOP　L

2. **下列程序是将** X、Y **双字变量的内容相加，其结果存放在** Z **变量中。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
X       DD      76543210H
Y       DD      251456865H
Z       DB      5 DUP（0）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，4
        _____(1)_____
        AND     AX，AX
L：     MOV     AL，BYTE  PTR  X[SI]
        _____(2)_____
        MOV     Z[SI]，AL
        INC     SI
        _____(3)_____
        MOV     AL，0
        _____(4)_____
        MOV     Z[SI]，AL
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）MOV　SI，0

（2）ADC　AL，BYTE PTR Y[SI]

（3）LOOP　L

（4）ADC　AL，0

3. **下列程序是将** FIR **的多位非压缩型** BCD **码与** SEC **的一位非压缩型** BCD **码相乘，其结果存放在** THIR **缓冲区中。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
FIR     DB      03H，08H，03H，…，08H
C       =       $-FIR
SEC     DB      07H
THIR    DB      C+1 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，C
        MOV     SI，0
        ____(1)____
L：      MOV     AL，FIR[SI]
        ____(2)____
        AAM
        ADD     AL，DH
        ____(3)____
        MOV     DH，AH
        MOV     THIR[SI]，AL
        ____(4)____
        LOOP    L
        MOV     THIR[SI]，AH
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）MOV　DH，0

（2）MUL　SEC

（3）AAA

（4）INC　SI

4. **下列程序是将键盘输入的十进制数（**-32768～32767**）转换为二进制数并送** BIN **保存。请在程序的空格处填写适当的指令。**

```
STACK   SEGMENT
        DB      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DB      10，0，10 DUP (0)
BIN     DW      0
```

```
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     DX，BUF
        MOV     AH，10
        INT     21H
              (1)
        MOV     CH，0
        LEA     SI，BUF+2
        CMP     BYTE  PTR [SI],‘－’
        PUSHF
              (2)
        INC     SI
        DEC     CX
        JMP     SIN
Q：      CMP     BYTE  PTR[SI],‘+’
        JNZ     SIN
        INC     SI
        DEC     CX
SIN：    MOV     AX，0
L：      MOV     DX，10
        MUL     DX
        AND     BYTE PTR[SI]，0FH
        ADD     AL，[SI]
              (3)
        INC     SI
        LOOP    L
        POPF
        JNZ     P
              (4)
P：      MOV     BIN，AX
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）MOV　CL，BUF+1

（2）JNZ　Q　**或**　JNE　Q

（3）ADC　AH，0

（4）NEG　AX

5. **下列程序是将** BUF **中的** 16 **位无符号二进制数转换为十进制数并输出。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
BUF     DW      74A5H
OUBUF   DB      6 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        LEA     BX, BUF+5
        MOV     BYTE PTR[BX],'$'
        DEC     BX
        ____(1)____
        MOV     CX, 10
L:      MOV     DX, 0
        ____(2)____
        ADD     DL, 30H
        ____(3)____
        DEC     BX
        OR      AX, AX
        JNZ     L
        ____(4)____
        MOV     DX, BX
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】(1) MOV　AX，BUF

(2) DIV　CX

(3) MOV　[BX]，DL

(4) INC　BX

6. **下列程序是检查并统计** BUF **缓冲区中以回车**（0DH）**结束的** ASCII **码串中十进制数的字符个数。若全部是十进制数字符，则将统计的结果送** RESUL **单元；否则输出**‘ERROR!’**。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
BUF     DB      '74A56796959786',0DH
RESUL   DW      0
ER      DB      'ERROR!$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
```

```
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AX，0
         LEA     BX，BUF
L：      CMP     BYTE  PTR[BX]，0DH
         ________(1)________
         CMP     BYTE  PTR[BX]，30H
         JB      ERR
         ________(2)________
         JA      ERR
         INC     AX
         INC     BX
         ________(3)________
DONE：   MOV     RESUL，AX
RE：     MOV     AH，4CH
         INT     21H
ERR：    LEA     DX，ER
         MOV     AH，9
         INT     21H
         ________(4)________
CODE     ENDS
         END     START
```

【解答】（1）JE　DONE　**或**　JZ　DONE

（2）CMP　BYTE　PTR[BX]，39H

（3）JMP　L

（4）JMP　RE

7. **下列程序是将** FIR **变量的多位压缩** BCD **码与** SEC **变量的两位压缩** BCD **码相乘，其结果送** THIR**。请在程序的空格处填写适当的指令。**

```
DATA     SEGMENT
FIR      DB      56H，90H，43H，…，28H
C        =       $-FIR
SEC      DB      67H
THIR     DB      C+1 DUP（0）
         DATA    ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     SI，0
         MOV     THIR[SI]，BYTE PTR 0
         ________(1)________
L：      MOV     BL，SEC
```

```
        MOV     AX, 0
P：     ADD     AL, FIR[SI]
            (2)
        XCHG    AH, AL
        ADC     AL, 0
        DAA
        XCHG    AH, AL
        XCHG    AL, BL
        SUB     AL, 1
            (3)
        XCHG    AL, BL
        JNZ     P
        ADD     AL, THIR[SI]
        DAA
        MOV     THIR[SI], AL
        XCHG    AH, AL
        ADC     AL, 0
        DAA
            (4)
        MOV     THIR[SI], AL
        LOOP    L
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】(1) MOV　CX，C

(2) DAA

(3) DAS

(4) INC　SI

8. **下列程序是统计 BUF 字缓冲区中正数、负数和 0 的个数并分别送 P、N 和 Z 字变量。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
BUF     DW      56H, 90H, 43H, …, 28H
C       =       $-FIR
P       DW      0
N       DW      0
Z       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START： MOV     AX, DATA
        MOV     DS, AX
```

```
        MOV     CX，C
        LEA     SI，BUF
                (1)
        CMP     AX，0
        JZ      ZR
                (2)
        INC     P
        JMP     NEXT
NE：    INC     N
        JMP     NEXT
ZR：    INC     Z
NEXT：          (3)
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）L：　MOV　AX，[SI]

（2）JS　NE

（3）ADD　SI，2

9. **下列程序是统计以** 0 **为结尾的** STR **字符串中大写字母、小写字母、数字和非数字字母的个数分别送** A、B、C、D **字单元保存。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
STR     DB      ‘ZSRU67797T?。，；，0PP；JO78545dwrwERqwe’，0
A       DW      0
B       DW      0
C       DW      0
D       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：MOV      AX，DATA
        MOV     DS，AX
        LEA     SI，STR
L：     MOV     AL，[SI]
        CMP     AL，0
        JZ      EXIT
        CMP     AL，30H
                (1)
        CMP     AL，39H
        JA      NEXT1
        INC     C
```

```
        JMP     CON
NEXT1：  CMP     AL，41H
        JB      NCHAR
        CMP     AL，5AH
                (2)
        INC     A
        JMP     CON
NEXT2：  CMP     AL，61H
        JB      NCHAR
                (3)
        JAE     NCHAR
        INC     B
        JMP     CON
NCHAR：  INC     D
CON：    INC     SI
                (4)
EXIT：   MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】（1）JB　　NCHAR

（2）JA　　NEXT2

（3）CMP　　AL，7BH

（4）JMP　　L

10. **下列程序是将** BUF **缓冲区的多精度数求补。请在程序的空格处填写适当的指令。**

```
STACK   SEGMENT
DB      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DB      13H，5DH，0AFH，34H，68，06H
COUNT   =       $-BUF
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
        MOV     CX，COUNT
        PUSH    CX
L1：             (1)
        INC     SI
                (2)
```

```
        POP     CX
        LEA     SI, BUF
                (3)
L2：    ADC     BYTE PTR[SI], 0
        INC     SI
                (4)
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】(1) NOT BYTE PTR[SI]

(2) LOOP L1

(3) STC

(4) LOOP L2

11. **下列程序是将** BCD **字节缓冲区中的数据转换为十六进制数并输出。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
BCD     DB      13H, 5DH, 0AFH, 34H, 68, 06H
COUNT   =       $-BCD
CRLF    DB      0DH, 0AH, 24H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START： MOV     AX, DATA
        MOV     DS, AX
        LEA     SI, BCD
                (1)
L：     MOV     DL, [SI]
        MOV     CL, 4
        SHR     DL, CL
        CMP     DL, 9
        JBE     N1
        ADD     DL, 7
N1：    ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     DL, [SI]
                (2)
        CMP     DL, 9
        JBE     N2
        ADD     DL, 7
N2：    ADD     DL, 30H
```

```
        MOV     AH，2
        INT     21H
        DEC     BX
                (3)
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】(1) MOV　BX，COUNT
(2) AND　DL，0FH
(3) JNZ　L

12. **下列程序是将有符号数字节缓冲区** BUF **中小于** 100 **的数送** BUF1 **字节缓冲区中保存。请在程序的空格处填写适当的指令。**

```
DATA    SEGMENT
BUF     DB      0A3H，8DH，0AFH，30H，68，088H
C       =       $-BUF
BUF1    DB      C DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，BUF
                (1)
        MOV     CX，C
L：      MOV     AL，[SI]
        CMP     AL，100
                (2)
        MOV     [DI]，AL
                (3)
NEXT：   INC     SI
        LOOP    L
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

【解答】(1) LEA　DI，BUF1　**或**　MOV　DI，OFFSET BUF1
(2) JGE　NEXT
(3) INC　DI

6.4.6 程序设计题

1. 将 BUF 缓冲区中的 100 个字的有符号数按正数和负数分开，并分别送至同一个数据段的 BUF1 和 BUF2 两个缓冲区中。

【解答】
```
DATA    SEGMENT
BUF     DW      100 DUP (?)
BUF1DW  100 DUP (0)
BUF2DW  100 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, 100
        LEA     SI, BUF
        LEA     DI, BUF1
        LEA     BX, BUF2
L:      MOV     AX, [SI]
        CMP     AX, 0
        JL      N
        MOV     [DI], AX
        ADD     DI, 2
        JMP     NEXT
N:      MOV     [BX], AX
        ADD     BX, 2
NEXT:   ADD     SI, 2
        LOOP    L
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

2. 编写一程序，统计 BUF 字缓冲区中的 255 个数据中为 0 的个数，并将统计的结果以十六进制数的形式显示输出。

【解答】
```
DATA    SEGMENT
BUF     DW      255 DUP (?)
ZR      DB      'ZRROR=$'
Z       DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
```

```
        MOV     CX, 255
        LEA     SI, BUF
L:      MOV     AX, [SI]
        CMP     AX, 0
        JNZ     NEXT
        INC     Z
NEXT:   ADD     SI, 2
        LOOP    L
        LEA     DX, ZR
        MOV     AH, 9
        INT     21H
        MOV     DL, Z
        MOV     CL, 4
        SHR     DL, CL
        CMP     DL, 9
        JBE     NEXT1
        ADD     DL, 7
NEXT1:  ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     DL, Z
        AND     DL, 0FH
        CMP     DL, 9
        JBE     NEXT2
        ADD     DL, 7
NEXT2:  ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

3. **有若干行字符串存放在以** BUF **为首地址的字节存储区中，最后以** 1AH **作结束标志。现需删除第** 4 **行的内容，并将删除后** BUF **缓冲区的内容显示输出。**

【解答】

```
DATA    SEGMENT
BUF     DB      'LINE1  AAAAAAAAAAA', 0DH, 0AH
        DB      'LINE2  BBBBBBBBBBBB', 0DH, 0AH
        DB      'LINE3  CCCCCCCCCCCC', 0DH, 0AH
        DB      'LINE4  DDDDDDDDDDD', 0DH, 0AH
        DB      'LINE5  EEEEEEEEEEEEE', 0DH, 0AH
        DB      'LINE6  FFFFFFFFFFFFFF', 0DH, 0AH, 1AH
N       =       4
DATA    ENDS
```

```
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，3
        LEA     SI，BUF-1
L1：     INC     SI
        CMP     [SI]，BYTE PTR 0AH
        JNZ     L1
        LOOP    L1
        MOC     DI，SI
L2：     INC     SI
        CMP     [SI]，BYTE PTR 0AH
        JNZ     L2
L3：     INC     SI
        INC     DI
        MOV     AL，[SI]
        MOV     [DI]，AL
        CMP     AL，1AH
        JNZ     L3
        LEA     SI，BUF
PLAY：   MOV     DL，[SI]
        CMP     DL，1AH
        JZ      EXIT
        MOV     AH，2
        INT     21H
        INC     SI
        JMP     PLAY
EXIT：   MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

4. **在** STR **字符串中搜索字符** A。**如果找到该字符，则用字符** B **替代字符** A；**如果未找到该字符，则在** STR **字符串后插入字符** B，**试编写此程序。**

【解答】

```
DATA    SEGMENT
STR     DB      ZSDRT5B6AABET45YDB6MM。/；L8’
C       =       $-STR
FLGE    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
```

```
        MOV     CX, C
        LEA     SI, STR
        MOV     FLGE, 0
L:      MOV     AL, [SI]
        CMP     AL,'A'
        JNZ     NEXT
        MOV     [SI], BYTE PTR 'B'
        MOV     FLGE, 1
NEXT:   INC     SI
        LOOP    L
        CMP     FLGE, 0
        JNZ     EXIT
        MOV     [SI], BYTE PTR 'B'
EXIT:   MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

5. **试编写一程序，要求比较两个字符串** STR1 **和** STR2 **中所含字符是否相同，若相同则显示输出‘** MATCH **’；若不相同则显示输出‘** NO MATCH **’。**

【解答】

```
DATA    SEGMENT
STR1    DB      'ZSDRT5B6AABET45YDB6MM。/;L8'
C1      =       $-STR1
STR2    DB      'ZSDRT5B6AABET45YDB6MM。/;L'
C2      =       $-STR2
N       DB      'NO  MATCH!$'
Y       DB      'MATCH!$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, C1
        CMP     CX, C2
        JNZ     NOEQ
        LEA     SI, STR1
        LEA     DI, STR2
L:      MOV     AL, [SI]
        CMP     AL, [DI]
        JNZ     NOEQ
        INC     SI
        INC     DI
        LOOP    L
        LEA     DX, Y
```

```
        MOV     AH，9
        INT     21H
        JMP     EXIT
NOEQ：  LEA     DX，N
        MOV     AH，9
        INT     21H
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

6. **设A和B缓冲区中是两个以若干个文本行组成的文本文件（文本文件以1AH结束），每个文本行以回车符（0DH）和换行符（0AH）结束。试编写一程序，以行为单位依次比较，如果两行相等，则往下比较；如果两行不相等，则显示这两行的行号和内容。**

【解答】

```
DATA    SEGMENT
A       DB      ‘MOV   AL，AH’，0DH，0AH
        DB      ‘SHR   AL，CL’，0DH，0AH
        DB      ‘ADD   AL，BL’，0DH，0AH，1AH
B       DB      ‘MOV   AL，AH’，0DH，0AH
        DB      ‘SHR   AL，CL’，0DH，0AH
        DB      ‘ADD   AL，DL’，0DH，0AH，1AH
N       DB      ‘第’，30H，‘行：’，0DH，0AH，‘$’
BUF1    DB      100 DUP（0）
BUF2    DB      100 DUP（0）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        LEA     SI，A
        LEA     DI，B
L：     MOV     AL，0
        MOV     CX，200
        LEA     BX，BUF1
L0：    MOV     [BX]，AL
        INC     BX
        LOOP    L0
        LEA     BX，BUF1
        MOV     DX，0
L1：    CMP     [SI]，BYTE PTR 1AH
        JZ      EXIT
        MOV     AL，[SI]
        MOV     [BX]，AL
        INC     SI
```

```
        INC     BX
        INC     DH
        CMP     AL, 0AH
        JNZ     L1
        MOV     [BX], BYTE PTR '$'
        LEA     BX, BUF2
L2:     CMP     [DI], BYTE PTR 1AH
        JZ      EXIT
        MOV     AL, [DI]
        MOV     [BX], AL
        INC     DI
        INC     BX
        INC     DL
        CMP     AL, 0AH
        JNZ     L2
        MOV     [BX], BYTE PTR '$'
        INC     N+2
        CMP     DH, DL
        JNZ     PLAY
        MOV     BX, 0
        MOV     CL, DL
        MOV     CH, 0
L3:     MOV     AL, BUF1[BX]
        CMP     AL, BUF2[BX]
        JNZ     PLAY
        INC     BX
        LOOP    L3
        JMP     L
PLAY:   LEA     DX, N
        MOV     AH, 9
        INT     21H
        LEA     DX, BUF1
        MOV     AH, 9
        INT     21H
        LEA     DX, BUF2
        MOV     AH, 9
        INT     21H
        JMP     L
EXIT:   MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

7. **试编写一程序，要求能从键盘接收一个个位数 N，然后响铃 N 次（响铃的 ASCII 码为 07）。**

【解答】
```
DATA    SEGMENT
BUF     DB      0
ERROR   DB      0DH, 0AH, 'INPUT ERROR!$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
L:      MOV     AH, 1
        INT     21H
        CMP     AL, 30H
        JB      ER
        CMP     AL, 39H
        JNA     CON
ER:     LEA     DX, ERROR
        MOV     AH, 9
        INT     21H
        JMP     L
CON:    MOV     BUF , AL
        AND     AL, 0FH
        CBW
        MOV     CX, AX
        JCXZ    EXIT
L:      MOV     DL, 7
        MOV     AH, 2
        INT     21H
        MOV     BX, 1000
DELAY:  DEC     BX
        JNZ     DELAY
        LOOP    L
EXIT:   MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

8. **试编写一个汇编语言程序，求出首地址为** ARY **的** 100 **个无符号字数组中的最小偶数，并把它存放在** BUF **中。**

【解答】
```
DATA    SEGMENT
        ARY     DW      100 DUP (?)
        BUF     DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
```

```
        MOV     DS, AX
        LEA     SI, ARY
        MOV     CX, 100
        MOV     AX, 0
L:      TEST    [SI], WORD PTR 1
        JNZ     N
        CMP     AX, [SI]
        JBE     N
        MOV     AX, [SI]
N:      ADD     SI, 2
        LOOP    L
        MOV     BUF, AX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

9. **设有一段英文，其字符变量名为** ENG**，并以＄字符结束。试编写一程序，查找单词** RED **在该文中出现的次数，并以“** RED **出现的次数：**X X X X**”的格式显示出次数。**

【解答】

```
DATA    SEGMENT
ENG     DB      'THIS IS RED。RED … $'
DISPL   DB      0DH, 0AH, 'RED 出现次数:'
DAT     DB      4 DUP ('  '), '$'
N       DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     N, 0
        LEA     SI, ENG
L:      CMP     [SI], BYTE PTR '$'
        JZ      DISP
        CMP     [SI], BYTE PTE 'R'
        JNZ     NEXT
        CMP     [SI+1], BYTE PTR 'E'
        JNZ     NEXT
        CMP     [SI+2], BYTE PTR 'D'
        JNZ     NEXT
        INC     N
        ADD     SI, 2
NEXT:   INC     SI
        JMP     L
DISP:   MOV     CX, 5
```

```
        LEA     SI, DAT+4
        MOV     AX, N
        MOV     BX, 10
DISP1:  MOV     DX, 0
        DIV     BX
        ADD     DL, 30H
        MOV     [SI], DL
        DEC     SI
        LOOP    DISP1
        LEA     DX, DISPL
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

10. **有一个首地址为 MEM 的 100 个字数组，试编制程序删除数组中所有为零的项，并将后续项向前移进，填补删除项，最后将数组的剩余部分以零补充。**

【解答】

```
DATA    SEGMENT
MEM     DW      100 DUP (?)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     SI, 198
        MOV     BX, -2
        MOV     CX, 100
COMP:   ADD     BX, 2
        CMP     MEM[BX], WORD PTR 0
        JZ      CONS
        LOOP    COMP
        JMP     EXIT
CONS:   MOV     DI, BX
CONS1:  CMP     DI, SI
        JNB     NOMOV
        MOV     AX, MEM[DI+2]
        MOV     MEM[DI], AX
        ADD     DI, 2
        JMP     CONS1
NOMOV:  MOV     WORD PTR[SI], 0
        LOOP    COMP
EXIT:   MOV     AH, 4CH
```

```
        INT     21H
CODE    ENDS
        END     START
```

11. **在 STR 到 STR＋99 个单元中存放着一个字符串，试编制一程序测试该字符串中是否含有数字。如有，则把 FLAG 标志置 1，否则将该标志清 0。**

【解答】

```
DATA    SEGMENT
STR     DB      100 DUP (?)
FLAG    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, 100
        MOV     SI, 0
L:      MOV     AL, STR[SI]
        CMP     AL, 30H
        JB      NNUM
        CMP     AL, 3AH
        JNB     NNUM
        MOV     FLAG, 1
        JMP     EXIT
NNUM:   INC     SI
        LOOP    L
        MOV     FLAG, 0
EXIT:   MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

12. **在首地址为 TABLE 的数组中按递增有序存放着 100H 个 16 位的补码数，试编写一个程序，把数组中出现次数最多的数及其出现次数分别存放在 NUMB 和 COUN 字单元中。**

【解答】

```
DATA    SEGMENT
TABLE   DW      100H DUP (?)
NUMB    DW      0
COUN    DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     CX, 100H
        MOV     SI, 0
```

```
        MOV     COUN，0
L0：    MOV     DX，0
        MOV     AX，TABLE[SI]
L：     CMP     AX，TABLE[SI]
        JNZ     DONE
        INC     DX
        ADD     SI，2
        LOOP    L
        CMP     DX，COUN
        JBE     EXIT
        MOV     COUN，DX
        MOV     NUMB，AX
        JMP     EXIT
DONE：  CMP     DX，COUN
        JBE     L0
        MOV     COUN，DX
        MOV     NUMB，AX
        JMP     L0
EXIT：  MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

13. **在首地址为** ARY **的字数组中，存放了** 100H **个** 16 **位的补码数，试编写一程序，求出它们的平均值并放在** V **字单元中；求出数组中有多少个数大于此平均值（整数部分），将结果放在** COUN **字单元中。**

【解答】

```
DATA    SEGMENT
ARY     DW      100H DUP（?）
V       DW      0，0
COUN    DW      0
ERROR   DB      ‘溢出错误！$’
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     CX，00FFH
        MOV     DX，0
        MOV     SI，0
        MOV     AX，ARY[SI]
        CWD
L：     ADD     SI，2
        ADD     AX，ARY[SI]
        ADC     DX，0
```

```
        JO      ER
        LOOP    L
        MOV     CX, 100H
        IDIV    CX
        MOV     V, AX
        MOV     V+2, DX
        MOV     SI, 0
COMP：  CMP     AX, ARY[SI]
        JLE     NIN
        INC     COUN
NIN：   ADD     SI, 2
        LOOP    COMP
        JMP     EXIT
ER：    LEA     DX, ERROR
        MOV     AH, 9
        INT     21H
EXIT：  MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

14. **试编制一个程序，把** BUF **缓冲区中的** 4 **位十六进制数转换为** ASCII **码，并将对应的** ASCII **码依次存放到** MEM **数组中的** 4 **个字节中(例如,当** BUF **缓冲区中的内容为** 2A49H **时，程序执行完后，**MEM **中的** 4 **个字节内容为** 39H、34H、41H **和** 32H**)，并将转换的结果输出。**

【解答】

```
DATA    SEGMENT
BUF     DW      5AD3H
MEM     DB      4 DUP (0)
DISP    DB      4 DUP (0),'$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA
START： MOV     AX, DATA
        MOV     DS, AX
        LEA     DI, MEM
        LEA     SI, DISP+3
        MOV     CH, 4
        MOV     CL, 4
        MOV     AX, BUF
L：     MOV     BX, AX
        AND     AX, 0FH
        CMP     AL, 9
        JA      CHAR
        ADD     AL, 7
```

```
CHAR：  ADD     AL，30H
        MOV     [DI]，AL
        MOV     [SI]，AL
        INC     DI
        DEC     SI
        MOV     AX，BX
        SHR     AX，CL
        DEC     CH
        JNZ     L
        LEA     DISP
        MOV     AH，9
        INT     21H
        MOV     AH，4CH
        INT     21H
CODE    ENDS
        END     START
```

15. **已知数组 A 包含 15 个互不相等的整数，数组 B 包含 20 个互不相等的整数，试编制一程序，把既在 A 中出现又在 B 中出现的整数存放于数组 C 中。**

【解答】

```
DATA    SEGMENT
A       DW      15 DUP (?)
B       DW      20 DUP (?)
C       DW      15 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     SI，0
        MOV     BX，0
        MOV     CX，15
L1：    PUSH    CX
        MOV     DI，0
        MOV     CX，20
        MOV     AX，A[SI]
L2：    CMP     AX，B[DI]
        JNZ     NO
        MOV     C[BX]，AX
        ADD     BX，2
NO：    ADD     DI，2
        LOOP    L2
        ADD     SI，2
        POP     CX
        LOOP    L1
```

```
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

第 7 章　子程序设计

子程序是一种较普遍和常用的基本程序结构形式之一。它是将功能通用的、使用频繁的一些程序段设计为子程序，可供程序中多处调用，以达到简化程序的目的。子程序又是模块化程序设计的基础，掌握好子程序的设计方法，有助于模块化程序的设计。

在本章的学习中，要求深刻理解子程序的概念、基本结构形式和作用；熟悉子程序文件说明的重要性及书写子程序文件说明的规定；掌握主程序和子程序的调用关系；熟记子程序调用和返回过程所完成的操作；深刻地理解子程序调用和返回的堆栈变化过程；熟练地掌握子程序设计的方法和技巧；灵活地运用各种子程序参数的传递方法设计程序；初步掌握子程序嵌套的程序设计方法；了解 DOS 系统功能调用的基本内容及调用方法。

【学习重点】

1. 子程序的设计。
2. 子程序的参数传递方法。

【学习难点】

子程序的嵌套和子程序的递归。

7.2.1　知识体系结构

在本章中，所讲述的知识体系结构如图 7.1 所示。

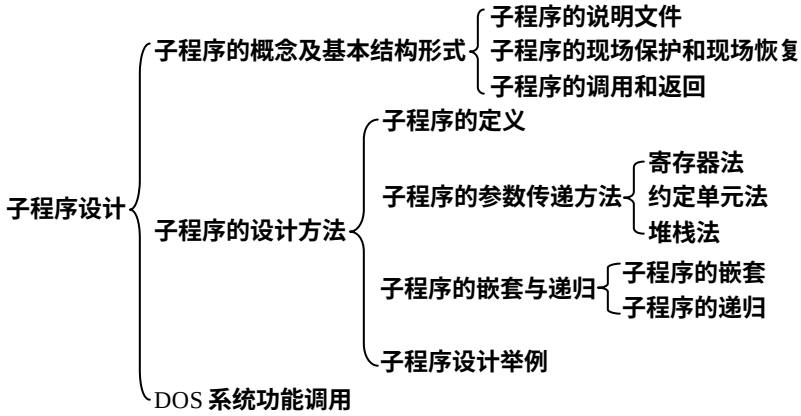


图 7.1　子程序设计的知识体系结构

7.2.2　知识点与考核要求

1. 子程序的概念及基本结构应达到“ 理解 ”的层次。

   （1）子程序的含义和作用，主程序和子程序的调用关系，子程序和过程语句的关系。

   （2）子程序文件说明的必要性及书写内容的规定。

2. 子程序的调用和返回应达到“ 简单应用 ”的层次。

   （1）子程序的调用和返回的操作过程。

   （2）子程序调用和返回时，入栈和出栈的具体操作内容及顺序，堆栈指针的变化情况。

3. 子程序设计应达到“ 综合应用 ”的层次。

   （1）子程序设计的方法和技巧。重点是主程序中的子程序调用指令的使用，定义过程语句，现场保护和现场恢复的程序设计。

   （2）能够分析主程序和子程序间调用和返回的程序流程，能够应用子程序设计方法编写程序。

4. 子程序的参数传递方法应达到“ 理解 ”的层次。

   （1）主程序和子程序间多种参数传递方法的操作原理。

   （2）主程序和子程序间多种参数传递方法的具体实现。

   （3）能够分析主程序和子程序间参数传递的程序实现。

5. 子程序的嵌套与递归应达到“ 理解 ”的层次。

   （1）嵌套子程序的含义及程序的流程。

   （2）递归子程序的含义和基本概念。

6. DOS 系统功能调用应达到“简单应用”的层次。

（1）DOS 系统功能调用的含义及常用的 DOS 系统功能调用。

（2）DOS 系统功能调用的方法，在编程时能够使用 DOS 系统功能调用。

7.3 例题分析

例 1　编写一子程序，将 AL 寄存器中的 1 位十六进制数转换成 ASCII 码显示输出。

解：要将 AL 寄存器中的 1 位十六进制数转换成 ASCII 码显示输出，这项工作应分两步来完成。第一步是将 1 位十六进制数转换成 ASCII 码，第二步是用 DOS 功能调用将 ASCII 码显示输出。在进行 ASCII 码转换的过程中，首先要区分这 1 位十六进制数是数字 0～9 还是字母 A～F。如果是数字 0～9，要将其转换成 ASCII 码，则应将其值加 30H；如果是字母 A～F，要将其转换成 ASCII 码，则应将其值加 37H。在进行显示输出时，只要将要显示字符的 ASCII 码送 DL 寄存器，然后进行功能调用就可以完成其操作。编写的程序清单如下：

```
HEXDIS      PROC    NEAR
            AND     AL，0FH
            ADD     AL，30H
            CMP     AL，39H
            JBE     NEXT
            ADD     AL，7
NEXT：      MOV     DL，AL
            MOV     AH，2
            INT     21H
            RET
HEXDIS      ENDP
```

在主程序调用子程序时，为了在子程序中能正确返回调用程序，一般说来，在一个子程序中至少要有一条返回指令。返回指令是子程序的出口，但返回指令不一定安排在子程序的最后。例如，将上例的程序改为如下的形式也是一样的。

```
HEXDIS      PROC    NEAR
            AND     AL，0FH
            CMP     AL，9
            JBE     NEXT
            ADD     AL，37H
            MOV     DL，AL
            MOV     AH，2
            INT     21H
            RET
NEXT：      ADD     AL，30H
```

```
            MOV     DL，AL
            MOV     AH，2
            INT     21H
            RET
HEXDIS      ENDP
```

例 2　编写一程序，将键盘输入的以非数字字符结束的十进制数转换成二进制数并依次送 BUF 字缓冲区保存，最后以输入回车符结束数据的输入。

解：在编写该程序时，要注意对输入字符的判断。如果是数字时，则应将输入的数字进行转换；如果是非数字字符（不是回车）时，则应结束数字的转换；如果是回车符时，则应结束数据的输入。在数据转换的过程中，可以采用如下的计算公式进行计算。

N=（…（((0）×10＋a1）×10＋a2）×10＋…）×10＋an

从表达式可以看出，如果将数据的初值设置为 0，每次将上次的数据乘以 10 再加本次输入的数据就可以转换成对应的二进制数。但应注意的是，在表达式中虽然是采用十进制数进行计算的，而计算机中是将这些十进制数用二进制数表示，并且以二进制数的形式进行计算，所以计算的结果是二进制数，这样就实现了十进制数转换为二进制数的目的。例如，输入的十进制数为 234，则表达式的计算为：

N=（((0）×10＋2）×10＋3）×10＋4

在（0）×10＋2 的计算中，虽然形式上是十进制数的计算，但在计算机中是按二进制数进行计算的，计算的结果为 00000010B。在进行第二步的计算过程中，其计算的方法如同第一步一样，其计算的操作是 00000010B×00001010B＋00000011 B=00010111B。在第三步的过程中，其计算的操作是 00010111B×00001010B＋00000100B=11101010B。

如果将调用程序和子程序编写在一个代码段中，则子程序的类型为 NEAR。编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DB      100  DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      30  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     SI，BUF
L：     MOV     DI，0
        CALL    DCB
        CMP     DI，0
```

```
        JNZ     EXIT
        MOV     [SI], BX
        ADD     SI, 2
        JMP     L
EXIT:   MOV     [SI], BX
        MOV     AH, 4CH
        INT     21H
;子程序名：DCB
;功能描述：将键盘输入的以非数字字符结束的十进制数转换成二进制数并送 BX 保存。
;入口参数：DI 寄存器存放无回车符输入的标志。
;出口参数：BX 寄存器存放键盘输入的十进制数转换成的二进制数。
;          DI 寄存器存放回车符输入的标志。
;其他说明：近程过程
DCB     PROC
        MOV     BX, 0
NEW:    MOV     AH, 1
        INT     21H
        CMP     AL, 0DH
        JNZ     NHC
        MOV     DI, 1
NDIG:   RET
NHC:    SUB     AL, 30H
        JL      NDIG
        CMP     AL, 9
        JG      NDIG
        CBW
        XCHG    AX, BX
        MOV     CX, 10
        MUL     CX
        XCHG    AX, BX
        ADD     BX, AX
        JMP     NEW
        DCB     ENDP
CODE    ENDS
        END     START
```

例 3　编写程序，其主程序和被调用的子程序在同一个代码段中。要求子程序的功能是将 BUF 中的 16 位无符号二进制数转换为 P 进制数，将 P 进制数的 ASCll 码按高位在前、低位在后（高位在低地址，低位在高地址）的顺序存放在 STR 字节缓冲区中，并将 STR 字节缓冲区中的字符显示输出。其中的 P 为 3～16 中的任一整数。

解：二进制数转换为 P 进制数可采用“除 P 取余”法，其算法如下：

将待转换的二进制数除以 P，得到第一个商数和第一个余数，这第一个余数就是所求 P 进制数的个位数；将第一个商数再除以 P，得到第二个商数和余数，这第二个余数就是所

求 P 进制数的十位数；……这一过程循环到商数为 0 时，所得到的余数就是所求 P 进制数的最高位。在定义 STR 字符串的长度时，由于 16 位二进制数转换成三进制数的位数最多需 12 位，为了能够利用 DOS 功能调用显示字符串，在数字转换的过程中，除了将转换的数字以 ASCII 码的形式送显示缓冲区外，还将回车符（0DH）、换行符（0AH）送显示缓冲区，最后送‘ $ ’到显示缓冲区，使该显示缓冲区外的字符串以‘ $ ’结束，所以显示字符串缓冲区最少应定义 15 个单元。在进行字符串显示输出的 DOS 功能调用时，就能将转换的数字以 P 进制数的形式显示输出。编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DW      100  DUP (0)
STACK   ENDS
DATA    SEGMENT
A       DW      1234H, 4352H, 3A62H, 6390H
N       =       ($-A) /2
BUF     DW      ?
STR     DB      15  DUP (?)
P       DW      ?
J       DB      ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        MOV     CX, N
        LEA     DI, A
L:      MOV     AX, [DI]      ;将被转换的二进制数送 BUF
        MOV     BUF, AX
        MOV     P, 8          ;将数制基数 8 送 P
        MOV     J,'Q'             ;将数制基数 8 的后缀 Q 送 J
        CALL    BCP
        MOV     P, 16             ;将数制基数 16 送 P
        MOV     J,'H'             ;将数制基数 16 的后缀 H 送 J
        CALL    BCP
        ADD     DI, 2
        LOOP    L
        MOV     AH, 4CH
        INT     21H
;子程序名：BCP
;功能描述：将 BUF 中的二进制数转换为 P 进制数并送 STR 保存。
;入口参数：BUF 缓冲区中存放待转换的二进制数，P 中存放待转换的数制基数，
;          J 中存放数制基数的后缀。
;出口参数：STR 缓冲区中存放转换的结果。
```

```
BCP     PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
        MOV     AX, BUF
        LEA     SI, STR
        MOV     BX, P
        MOV     CX, 0
L1：    MOV     DX, 0
        DIV     BX
        PUSH    DX
        INC     CX
        OR      AX, AX
        JNZ     L1
L2：    POP     AX
        CMP     AL, 10
        JB      L3
        ADD     AL, 7
L3：    ADD     AL, 30H
        MOV     [SI], AL
        INC     SI
        LOOP    L2
        MOV     AL, J
        MOV     [SI], AL
        MOV     [SI+1], 0DH
        MOV     [SI+2], 0AH
        MOV     [SI+3],‘ $ ’
        LEA     DX, STR
        MOV     AH, 9
        INT     21H
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
BCP     ENDP
CODE    ENDS
        END     START
```

例 4　编写一程序，其主程序和子程序在同一个代码段中。要求子程序将 DATA1 和 DATA2 字节缓冲区中的 N 个字节的内容相加，其结果存放在 SUM 字节缓冲区中。

解：这是一个多精度的计算程序。如果假定最高字节存放在高地址，最低字节存放在低地址，则应该从低地址单元中的内容开始进行相加。由于相加的结果在最高位可能会产生溢出，为了保证计算结果的正确性，所以应该将最高位的进位送到结果单元的最高位。如果采用约定单元法传递参数，编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DW          100  DUP (0)
STACK   ENDS
DATA    SEGMENT
DATA1   DB          10  DUP  (?)
DATA2   DB          10  DUP  (?)
SUM     DB          11  DUP  (0)
N       DW          10
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        CALL    MADD
        MOV     AH，4CH
        INT     21H
；子程序名：MADD
；功能描述：两个 N 字节的二进制数相加。
；入口参数：DATA1 和 DATA2 缓冲区中分别存放要相加的 N 个字节的二进制数。
；出口参数：SUM 缓冲区中存放结果。
；其他说明：  (1) N 字节的二进制数据的存放次序采用“低→高”的原则。
；            (2) 可能产生的进位存放在 SUM 开始的第 N+1 个字节中。
MADD    PROC
        PUSH    AX                  ；保护寄存器中的内容
        PUSH    CX
        PUSH    SI
        MOV     CX，N               ；置循环初值
        XOR     SI，SI
M1：     MOV     AL，DATA1 [SI]      ；DATA1 与 DATA2 相加
        ADC     AL，DATA2 [SI]
        MOV     SUM [SI]，AL
        INC     SI
        LOOP    M1
        MOV     AL，0               ；最高位的处理
        ADC     AL，0
        MOV     SUM[ SI ]，AL
        POP     SI                  ；恢复寄存器中的内容
```

```
        POP     CX
        POP     AX
        RET
MADD    ENDP
CODE    ENDS
        END     START
```

从以上的程序可以看出，参加运算的数必须先送到约定的存储单元，然后才能进行运算。如果要改变存储单元的地址，则这种约定单元法就无法完成。如果在调用程序中要增加另一次调用，将 BUF1 和 BUF2 中的 M 个字节的内容相加，其结果存放在 S 缓冲区中，则该子程序就不能完成其操作。所以，利用约定单元法直接传递参加运算的数据，通用性较差。特别是当传递的参数较多时，用约定单元法就更难保持良好的通用性。

为了解决子程序的通用性问题，通常采用地址表法传递参数。这种方法是把参数组成一张参数表存放在某个存储区中，然后只要主程序（调用程序）与子程序约定好这个存储区的首地址和存放的参数，在主程序中将参数传送给地址表，在子程序中根据地址表给定的参数进行运算，就可以完成主程序和子程序约定的功能。采用地址法传递参数，编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DW      100  DUP (0)
STACK   ENDS
DATA    SEGMENT
TAB     DW      3 DUP   (0)
A1      DB      10 DUP  (?)
B1      DB      10 DUP  (?)
S1      DB      11 DUP  (?)
C1      DW      10
A2      DB      30 DUP  (?)
B2      DB      30 DUP  (?)
S2      DB      31 DUP  (?)
C2      DW      30
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     AX，A1
        MOV     TAB，AX
        LEA     AX，B1
        MOV     TAB+2，AX
        LEA     AX，S1
        MOV     TAB+4，AX
```

```
        MOV     AX，C1
        MOV     TAB+6
        CALL    MBA
        LEA     AX，A2
        MOV     TAB，AX
        LEA     AX，B2
        MOV     TAB+2，AX
        LEA     AX，S2
        MOV     TAB+4，AX
        MOV     AX，C2
        MOV     TAB+6
        CALL    MBA
        MOV     AH，4CH
        INT     21H
；子程序名：MBA
；功能描述：多字节的二进制数相加。
；入口参数：TAB 表中存放二进制数缓冲区的首地址、相加个数地址、结果的地址。
；出口参数：二进制数缓冲区中的内容相加后，其和存放在结果地址单元。
MBA     PROC
        PUSH    AX                  ；保护寄存器中的内容
        PUSH    BX
        PUSH    CX
        PUSH    SI
        PUSH    DI
        MOV     SI，TAB             ；置循环初值
        MOV     DI，TAB+2
        MOV     BX，TAB+4
        MOV     CX，TAB+6
M1：    MOV     AL，[SI]
        ADC     AL，[DI]
        MOV     [BX]，AL
        INC     SI
        INC     DI
        INC     BX
        LOOP    M1
        MOV     AL，0               ；最高位的处理
        ADC     AL，0
        MOV     [BX]，AL
        POP     DI                  ；恢复寄存器中的内容
        POP     SI
        POP     CX
        POP     BX
        POP     AX
        RET
```

```
MBA     ENDP
CODE    ENDS
        END     START
```

例 5　利用堆栈传递参数的方法编写一子程序，实现主程序调用不同代码段的数组求和的子程序。

解：在数据段中定义两个数组 ARYA、ARYB。主程序和子程序分别安排在两个不同的代码段中，子程序应定义为 FAR 类型。主程序和子程序之间的参数传递是利用堆栈实现的。在程序设计时，一定要注意堆栈的变化，避免参数的读取和程序的返回出现错误。

编写的程序清单如下：

```
STACK   SEGMENT PARA    STACK  'STACK'
        DB      100  DUP (?)
STACK   ENDS
DATA    SEGMENT
ARYA    DB      a1, a2, a3, …, an        ;定义数组 A
NA      =       $-ARYA                   ;数组 A 的元素个数
SUMA    DW      ?                        ;数组 A 和数存放单元
ARYB    DB      b1, b2, b3, …, bn        ;定义数组 B
NB      =       $-ARYB                   ;数组 B 的元素个数
SUMA    DW      ?                        ;数组 B 和数存放单元
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        MOV     AX, NA                   ;数组 A 的元素个数压栈（参数 1）
        PUSH    AX
        LEA     AX, ARYA                 ;数组 A 的元素首地址压栈（参数 2）
        PUSH    AX
        LEA     AX, SUMA                 ;数组 A 元素的和数地址压栈（参数 3）
        PUSH    AX
        CALL    FAR  PTR SUM             ;此时，将断点地址 CS:IP 压栈保存
        MOV     AX, NB                   ;数组 B 的元素个数压栈（参数 1）
        PUSH    AX
        LEA     AX, ARYB                 ;数组 B 的元素首地址压栈（参数 2）
        PUSH    AX
        LEA     AX, SUMB                 ;数组 B 元素的和数地址压栈（参数 3）
        PUSH    AX
        CALL    FAR  PTR SUM             ;此时，将断点地址 CS:IP 压栈保存
        MOV     AH, 4CH
        INT     21H
```

```
CODE    ENDS
；子程序名：SUM
；功能描述：完成对数组元素的求和。
；入口参数：数组元素的个数、数组的首地址、和数的地址保存在主程序所定义
；          的堆栈存储单元中。
；出口参数：求得的数组元素之和保存在主程序所定义的数据区存储单元中。
PROCE   SEGMENT
        ASSUME  CS：PROCE，DS：DATA，SS：STACK
SUM     PROC    FAR
        PUSH    AX              ；保护寄存器的内容
        PUSH    BX
        PUSH    CX
        PUSH    BP
        PUSHF
        MOV     BP，SP          ；BP 指向栈顶
        MOV     CX，[BP+18]     ；取数组元素的个数（参数 1）
        MOV     BX，[BP+16]     ；取数组元素的首地址（参数 2）
        MOV     AX，0           ；AX 清 0
LOPI：  ADD     AL，[BX]        ；数组元素求和
        ADC     AH，0           ；低位字节进位时加至高位字节 AH 中
        INC     BX              ；修改地址
        LOOP    LOPI
        MOV     BX，[BP+14]     ；取数组 A 元素的和数地址（参数 3）
        MOV     [BX]，AX        ；送结果
        POPF                    ；恢复寄存器的内容
        POP     BP
        POP     CX
        POP     BX
        POP     AX
        RET     6
SUM     ENDP
PROCE   ENDS
        END     START
```

程序执行过程中，堆栈中内容的变化情况如图 7.2 所示。

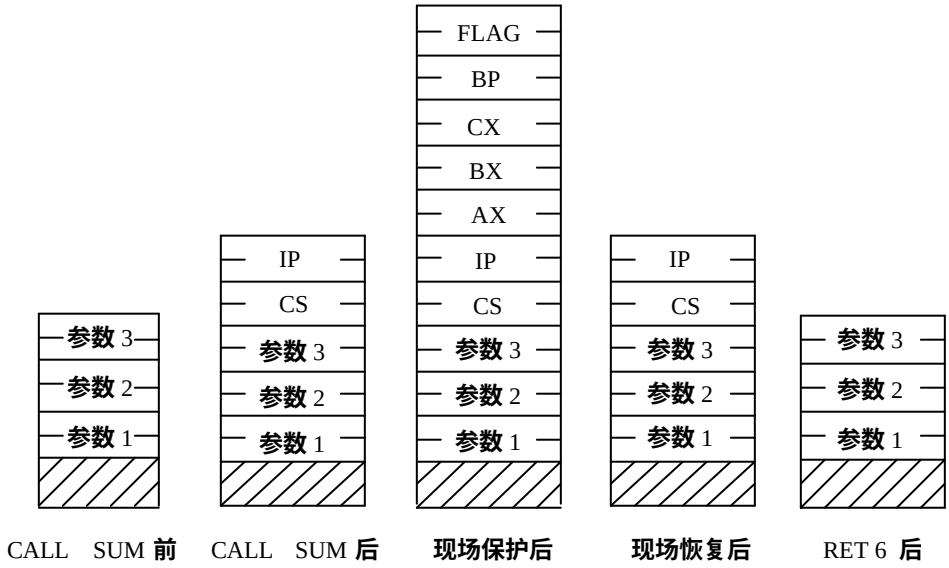


图 7.2　例 5 程序执行堆栈变化示意图

例 6　设 ARY1 和 ARY2 是两个长度都为 10 的双字数组。请用比例变量寻址方式编写一子程序，对两个数组中的对应元素分别相乘，其结果存放在 ARY3 数组中。

解：根据题目的要求，由于比例变量寻址方式只有在 386 及后继机型才有此类寻址方式，所以，必须用 386 及后继机型的指令才能完成该操作。编写的程序清单如下：

```
        .MODEL      SMALL
        .386
        .STACK      200H
        .DATA
ARY1    DD      10 DUP (?)
ARY2    DD      10 DUP (?)
ARY3    DQ      10 DUP (?)
TAB     DD      3 DUP (0)
        .CODE
START:  MOV     AX,@DATA
        MOV     DS,AX
        LEA     EAX,ARY1
        MOV     TAB,EAX
        LEA     EAX,ARY2
        MOV     TAB+4,EAX
        LEA     EAX,ARY3
        MOV     TAB+8,EAX
        CALL    AMUL
        MOV     AX,4C00H
        INT     21H
;子程序名：AMUL
```

```
;功能描述：两个数组中对应的 10 个元素分别相乘，其结果存放在第三个数组中。
;入口参数：TAB 表中分别存放着第一个数组元素的地址、第二个数组元素的地址和
;          第三个数组元素的地址。
;出口参数：两个数组中对应的 10 个元素分别相乘的结果存放在第三个数组中。
AMUL    PROC
        MOV     ESI, TAB
        MOV     EDI, TAB+4
        MOV     EBP, TAB+8
        MOV     EBX, 0
        MOV     CX, 10
L:      MOV     EAX, [ESI][EBX*4]
        MUL     [EDI][EBX*4]
        MOV     DWORD  PTR  [EBP][EBX*8], EAX
        MOV     DWORD  PTR  [EBP+4][EBX*8], EDX
        INC     EBX
        LOOP    L
        RET
AMUL    ENDP
        END     START
```

例 7　编写一程序，要求将从键盘输入的 0～FFFFH 的十六进制正整数转换为十进制数，并将转换的结果显示输出。

解：如果要完成题目的要求，可以把这一程序分解成一个主程序、一个从键盘输入的十六进制正整数转换为二进制数的子程序 HDC、一个二进制正整数转换为十进制数的子程序 BDC、一个十进制数转换为 ASCII 码显示输出的子程序 DISP 和显示输出换行符的子程序 CRLF。其中主程序可调用子程序 HDC、BDC、CRLF，在 BDC 子程序中可再次调用 DISP 子程序，实现子程序的嵌套调用。编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DW      100  DUP (0)
STACK   ENDS
DATA    SEGMENT
STR     DB          '是否继续输入数据？需继续输入请按 Y 键，否则按其他键！$'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
L:      CALL    HDC
        CALL    CRLF
        CALL    BDC
        CALL    CRLF
```

```
        LEA     DX, STR
        MOV     AH, 9
        INT     21H
        MOV     AH, 1
        INT     21H
        CMP     AL,‘Y’
        JZ      L
        CMP     AL,‘y’
        JZ      L
        MOV     AH, 4CH
        INT     21H
```

；子程序名：HDC
；功能描述：将从键盘输入的十六进制正整数转换为二进制数并送 BX**。**
；入口参数：无
；出口参数：BX **寄存器存放键盘输入的十六进制数转换后得到的二进制数。**

```
HDC     PROC
        PUSH    AX
        PUSH    CX
        MOV     BX, 0
NKEY:   MOV     AH, 1
        INT     21H
        CMP     AL, 30H
        JL      EXIT
        CMP     AL, 3AH
        JL      DIGIT
        CMP     AL, 41H
        JL      EXIT
        CMP     AL, 47H
        JL      CHAR
        CMP     AL, 61H
        JL      EXIT
        CMP     AL, 67H
        JGE     EXIT
        SUB     AL, 20H
CHAR:   SUB     AL, 7
DIGIT:  SUB     AL, 30H
        MOV     CL, 4
        SHL     BX, CL
        MOV     AH, 0
        ADD     BX, AX
        JMP     NKEY
        POP     CX
        POP     AX
EXIT:   RET
```

```
HDC     ENDP
；子程序名：BDC
；功能描述：将 BX 寄存器中的二进制正整数转换为十进制数显示输出。
；入口参数：BX 寄存器中存放二进制数。
；出口参数：十进制数显示输出。
BDC     PROC
        PUSH    CX
        MOV     CX，10000
        CALL    DISP
        MOV     CX，1000
        CALL    DISP
        MOV     CX，100
        CALL    DISP
        MOV     CX，10
        CALL    DISP
        MOV     CX，1
        CALL    DISP
        POP     CX
        RET
BDC     ENDP
；子程序名：DISP
；功能描述：将 BX 寄存器中的二进制数除以 CX 寄存器中的值，并显示输出商。
；入口参数：BX 寄存器中存放二进制数被除数。
；          CX 寄存器中存放二进制数除数。
；出口参数：BX 寄存器中存放 BX/CX 后的二进制余数。
DISP    PROC
        PUSH    AX
        PUSH    CX
        PUSH    DX
        MOV     AX，BX
        MOV     DX，0
        DIV     CX
        MOV     BX，DX
        MOV     DL，AL
        ADD     DL，30H
        MOV     AH，2
        INT     21H
        POP     DX
        POP     CX
        POP     AX
        RET
DISP    ENDP
；子程序名：CRLF
；功能描述：显示输出回车换行。
```

```
;入口参数：无
;出口参数：无
CRLF    PROC
        PUSH    AX
        PUSH    DX
        MOV     DL，0DH
        MOV     AH，2
        INT     21H
        MOV     DL，0AH
        MOV     AH，2
        INT     21H
        POP     DX
        POP     AX
        RET
CRLF    ENDP
CODE    ENDS
        END     START
```

例 8　要求编写计算 N！（N≥0）的程序并将计算结果按十进制数显示输出。

解：该程序分两步，第一步计算 N！，第二步将二进制数转换为十进制数并显示输出。在计算 N！时，需按如下表达式计算。

N！= N×（N-1）×（N-2）×…×2×l

其递归定义如下：

0！= 1

N！=　N×（N-l）！（N≥0）

假定，求 N！是一个子程序，那么，根据递归定义 N！=N×（N－1）！，在求 N！时，必须递归调用求 N！的子程序求出（N-1）！（可以将 N-1 看作一个值调用 N！，以下类推）；在求（N-1）！时，由于（N-1）！=（N-1）×（N-2）！，所以必须递归调用 N！的子程序求出（N-2）！；依此类推，直到求出 0！。然后，再反向将每次调用求出的结果与上次调用时的（N-1）相乘，直到求出 N！为止。

但每次调用所使用的参数都不相同，所以，递归子程序的设计必须保证每次调用都不破坏以前调用时所用的参数和中间结果。在每次调用前，一般把调用的参数、寄存器内容及所有的中间结果都存放在堆栈中。在返回时，将调用前压堆栈的参数、寄存器内容及所有的中间结果从堆栈中弹出，以保证能按反向次序退出并返回主程序。

编写的程序清单如下：

```
STACK   SEGMENT     STACK
        DW          100  DUP (0)
TOP     LABEL       WORD
STACK   ENDS
DATA    SEGMENT
```

```
RESUL   DW      0
N       DW      5
OUTPU   DB      'N!='
BUF     DB      5 DUP ('  '),0DH,0AH,'$'
CONT    DW      10000,1000,100,10,1
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:  MOV     AX,DATA
        MOV     DS,AX
        MOV     AX,STACK
        MOV     SS,AX
        MOV     SP,OFFSET  TOP
        LEA     AX,RESUL
        PUSH    AX
        MOV     AX,N
        PUSH    AX
        CALL    FACT
        MOV     RESUL,AX
        CALL    C2T10
        LEA     DX,OUTPU
        MOV     AH,9
        INT     21H
        MOV     AH,4CH
        INT     21H
;子程序名:FACT
;功能描述:求N!。
;入口参数:将结果的地址RESUL和N值压入堆栈。
;出口参数:AX中存放求出的N!的结果。
FACT    PROC
        PUSH    BP
        MOV     BP,SP
        MOV     BX,[BP+6]
        MOV     AX,[BP+4]
        CMP     AX,0
        JZ      DONE
        PUSH    BX
        DEC     AX
        PUSH    AX
        CALL    FACT
        MOV     AX,[BP-2]
        MUL     WORD  PTR [BP+4]
        JMP     RETU
DONE:   MOV     AX,1
```

```
RETU：  MOV     [BP+6], AX
        POP     BP
        RET     4
FACT    ENDP
```

；子程序名：C2T10

；功能描述：将 RESUL **中的二进制数转换为十进制数并存放在** BUF **缓冲区中。**

；入口参数：RESUL **中存放二进制数。**

；出口参数：BUF **缓冲区中存放转换后的十进制数。**

```
C2T10   PROC
        MOV     AX, RESUL
        MOV     SI, 0
        MOV     CX, 5
        LEA     BX, CONT
CVDL：  CWD
        DIV     WORD  PTR [BX]
        ADD     AL, 30H
        MOV     BUF[SI], AL
        INC     SI
        MOV     AX, DX
        ADD     BX, 2
        LOOP    CVDL
        RET
C2T10   ENDP
CODE    ENDS
        END     START
```

在程序运行的过程中，FACT **不断调用自身，每调用一次就将（**N-l**）及有关信息入栈，直到** N **等于基数** 0 **为止。当程序运行到** N **等于** 0 **时开始返回，程序在不断返回的过程中，每次执行** N×（N-1）**!，并保存中间结果直到** N **等于给定值为止。在程序运行过程中，每次调用时在堆栈中保留一次调用的中间结果，程序返回时，则每返回一次就从堆栈中取出一个保留的中间结果进行计算。当程序运行完毕后，堆栈恢复原状，在** RESUL **单元中得到计算结果。当** N=5 **时，计算结果** RESUL **单元中的内容为** 120 **当** N=3 **时，计算结果** RESUL **单元中的内容为** 6**。当** N=3 **时的堆栈状态如图** 7.3 **所示。为了方便起见，图中的每一格表示两个单元。**

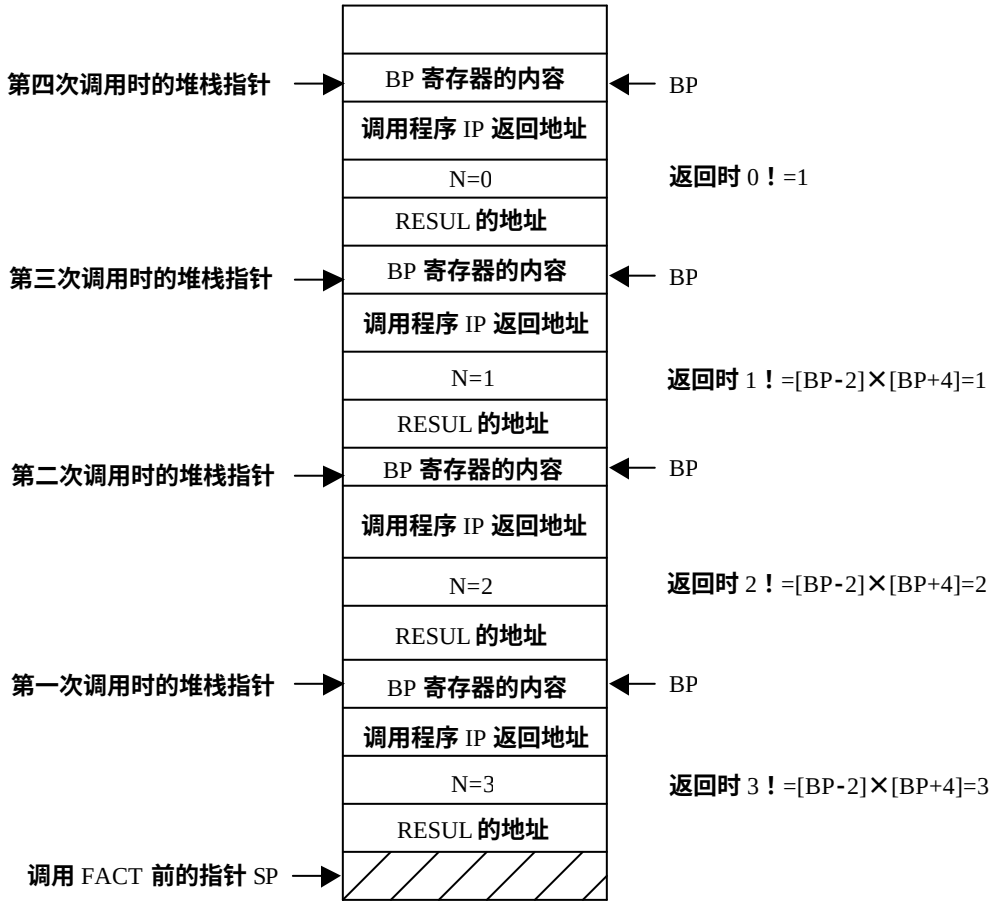


图 7.3　程序执行 N=3 时的堆栈变化状态

7.4 练习题与参考答案

7.4.1 单项选择题

1. 在子程序中，如果利用堆栈保护现场，在恢复现场时，先压堆栈的内容（　　）。

   A. 先弹堆栈　　B. 后弹堆栈　　C. 不分先后弹堆栈　　D. 任意弹堆栈

2. 下列叙述不正确的是（　　）。

   A. 在子程序中的现场保护只能用堆栈来实现
   B. 在子程序中的现场保护用堆栈来实现是其中的一种方法
   C. 在子程序中的现场保护可以有多种实现方法
   D. 在子程序中的现场保护可以将要保护的内容送内存变量来实现

3. 下列定义子程序的是（　　）。

A.
```
PNAME   PROC
        …
PNAME   ENDS
```
B.
```
PNAME   PROC
        …
        ENDM
```
C.
```
PNAME   PROC
        …
PNAME   ENDP
```
D.
```
PNAME   PROC
        …
PNAME   END
```

4. 子程序重定位可选在内存中的整个区域内，在进行程序设计时，子程序应采用（　　）。

A. 绝对地址　　B. 相对地址　　C. 逻辑地址　　D. 物理地址

5. 下列叙述不正确的是（　　）。

A. 在子程序中可以再次调用子程序
B. 在主程序中一般用跳转指令转移到子程序
C. 在子程序中一般用返回指令返回到主程序
D. 在主程序中一般用调用指令转移到子程序

6. 下列叙述正确的是（　　）。

A. 执行一条段内返回指令，先从堆栈弹两个字节的内容，然后 SP 指针减 2
B. 执行一条段内返回指令，先 SP 指针减 2，然后从堆栈弹两个字节的内容
C. 执行一条段内返回指令，先 SP 指针加 2，然后从堆栈弹两个字节的内容
D. 执行一条段内返回指令，先从堆栈弹两个字节的内容，然后 SP 指针加 2

7. 下列叙述正确的是（　　）。

A. 执行一条段间调用指令，先将 CS、IP 的内容压栈，然后将目的地址送 CS 和 IP
B. 执行一条段间调用指令，先将目的地址送 CS 和 IP，然后将 CS、IP 的内容压栈
C. 执行一条段间调用指令，先将 CS、IP 的内容压栈，然后 SP 指针加 2
D. 执行一条段间调用指令，先将 CS、IP 的内容压栈，然后 SP 指针减 2

8. 下列叙述不正确的是（　　）。

A. 在汇编语言程序中，每一个过程允许有多条 RET 指令
B. 在汇编语言程序中，每一个过程只允许出现一条 RET 指令
C. 在汇编语言程序中，每一个过程结束之前一定有一条 RET 指令
D. 在汇编语言程序中，以过程形式表示的代码段一定有一条 RET 指令存在

9. 下列叙述中属于子程序嵌套的是（　　）。

A. 主程序调用子程序 1　　B. 主程序调用子程序 2
C. 主程序调用子程序 3　　D. 子程序 1 调用子程序 2

10. 下列叙述中属于子程序直接递归调用的是（　　）。

A. 子程序 1 调用子程序 2　　B. 子程序 2 调用子程序 3
C. 子程序 3 调用子程序 4　　D. 子程序 4 调用子程序 4

11. 在进行 DOS 功能调用前，其功能号应先送（　　）。

A. AH 寄存器　　B. BH 寄存器　　C. CH 寄存器　　D. DH 寄存器

12. 执行“ INT　10H ”指令时，中断向量地址是（　　）。

A. 10H　　B. 20H　　C. 30H　　D. 40H

13. 执行“ INT　10H ”指令时，其中断处理程序的入口地址存放在（　　）。

A. 10H～13H　　B. 20H～23H　　C. 30H～33H　　D. 40H～43H

14. 执行“ INT　10H ”指令时，压堆栈的内容有（　　）。

A. PSW、CS、IP　　B. CS、IP　　C. PSW、IP　　D. PSW、CS

15. 在进行字符串显示的 DOS 功能调用时，要求字符串的最后一个字符是（　　）。

A. 0　　B. ‘ 0 ’　　C. ‘ $ ’　　D. 0DH

单项选择题参考答案

1. B	2. A	3. C	4. B	5. B	6. D
7. A	8. B	9. D	10. D	11. A	12. D
13. D	14. A	15. C			

7.4.2　多项选择题

1. 一般的子程序说明文件有（　　）。

A. 子程序名　　B. 功能描述　　C. 入口参数　　D. 出口参数

2. 在子程序的设计中，通常采用现场保护和现场恢复的方法有（　　）。

A. 寄存器法　　B. 堆栈法　　C. 约定单元法　　D. 变元法

3. 子程序的参数传递方法有（　　）。

A. 寄存器法　　B. 约定单元法　　C. 堆栈法　　D. 变元法

4. 下列叙述中属于子程序嵌套的有（　　）。

A. 子程序 1 调用子程序 2，子程序 2 调用子程序 3
B. 主程序调用子程序 2
C. 主程序调用子程序 3

D. 子程序 3 调用子程序 4

5. 下列叙述中属于子程序递归调用的有（　　）。

A. 子程序 1 调用子程序 2，子程序 2 调用子程序 3
B. 子程序 2 调用子程序 3，子程序 3 调用子程序 2
C. 子程序 3 调用子程序 4，子程序 4 调用子程序 3
D. 子程序 3 调用子程序 3

6. 在执行“CALL　FAR　PTR　M”指令后，完成的操作有（　　）。

A. CS、IP 入栈　　B. SP+2　　C. SP+4　　D. 目的地址送 CS、IP

7. 下列叙述正确的有（　　）。

A. 如果主程序和子程序 A 在同一代码段，则用“CALL　A”可以实现子程序的调用
B. 如果主程序和子程序 A 不在同一代码段，则用“CALL A”可实现子程序的调用
C. 如果主程序和子程序 A 在同一代码段，则用“CALL　FAR　PTR　A”可以实现子程序的调用
D. 如果主程序和子程序 A 不在同一代码段，则用“CALL　FAR　PTR　A”可以实现子程序的调用

8. 下列叙述不正确的有（　　）。

A. 在子程序设计中，不论采用什么参数传递方法，必须将所有寄存器的内容保护起来
B. 在子程序设计中，为了防止寄存器内容被破坏，通常将有关寄存器的内容保护起来
C. 在子程序设计中，都是采用寄存器法传递参数
D. 在子程序设计中，子程序都只安排一个出口

9. 在进行字符串输入的 DOS 功能调用时，输入字符串缓冲区中的内容是（　　）。

A. 第一个单元存放字符个数
B. 第二个单元存放的是实际输入字符的个数
C. 第三个单元开始存放输入字符的 ASCII 码
D. 最后一个单元存放回车符

10. 如果要将 BUF 字符缓冲区的内容显示输出，在采用 DOS 功能调用前，必须（　　）。

A. 将 BUF 字符缓冲区的段地址送 DS
B. 将 BUF 字符缓冲区的段地址送 ES
C. 将 BUF 字符缓冲区的偏移地址送 DX
D. 将功能号 9 送 AH 寄存器

多项选择题参考答案

1. ABCD	2. BC	3. ABC	4. AD	5. BCD
6. ACD	7. ACD	8. ACD	9. BC	10. ACD

7.4.3　填空题

1. 调用子程序通常用_____指令，返回调用程序通常用_____指令。

答：CALL（调用）　RET（返回）

2. 在子程序的设计中，通常用堆栈来保护现场和恢复现场。而堆栈的操作原则是_____。

答：后进先出（先进后出）

3. 如果主程序和子程序在同一个代码段，则主程序调用子程序时只改变 _____地址。如果主程序和子程序不在同一个代码段，则主程序调用子程序时要改变 _____地址。

答：IP（偏移）　CS 和 IP（段地址和偏移地址）

4. 在调用程序中调用子程序的 CALL 指令执行后，压堆栈的内容是_____指令的下一条指令的地址。在子程序中的 RET 指令将返回_____ 指令以继续执行程序。

答：CALL　CALL 指令的下一条

5. 子程序（过程）的定义是以“过程名 _____ ”开始，以“过程名 _____ ”结束。

答：PROC　[NEAR/FAR]，ENDP

6. 在调用子程序前，如果将要参加运算的数据送寄存器，这种参数传递方法称为 _____；如果将要参加运算的数据送指定的内存单元，这种参数传递方法称为 _____；如果将要参加运算的数据压堆栈，这种参数传递方法称为 _____。

答：寄存器法　约定单元法　堆栈法

7. 一个子程序调用另一个子程序称为 _____ ；一个子程序直接或间接调用该子程序本身称为_____ 。

答：子程序嵌套　递归调用

8. 以过程定义的子程序有两种类型的属性，它们分别是_____和_____ 。

答：NEAR（近程）　FAR（远程）

9. 在程序设计中，利用堆栈不仅可以保存_____ ，而且还可以保存主程序和子程序之间传递的参数，这些参数既可以是_____ ，也可以是 _____ 。

答：地址　数据（地址）　地址（数据）

10. 在程序的执行过程中，近程调用时 CALL 指令执行完后压堆栈的内容是_____个

字节，远程调用时 CALL 指令执行完后压堆栈的内容是_____个字节。

答：2　4

11. 在编制子程序时，_____的使用是十分频繁的，它不仅可以用来保存返回的地址，而且还可以用来存放主程序与子程序之间传递的_____ 。

答：堆栈　参数

12. 子程序的正确执行是由主程序中的_____指令和子程序的_____指令来完成的。

答：调用（CALL）　返回（RET）

13. 在进行“ INT　45H ”功能调用时，其中断类型号为_____，调用的功能号应送_____。

答：45H　AL 寄存器

14. DOS 系统功能调用是根据中断类型号从中断向量表中取中断处理程序的入口地址。每一个中断处理程序的入口地址占用内存_____个字节，其中的低地址字用来存放中断处理程序的_____，高地址字用来存放中断处理程序的_____。

答：4　偏移地址　段地址

15. 中断向量表占用内存_____字节，它的物理地址范围是_____。

答：1K　0～1023（0000～03FFH）

16. 在 DOS 系统功能调用中，01 号功能调用表示_____，它的出口参数在_____中。02 号功能调用表示_____，它的入口参数在_____中。

答：带显示的键盘输入　AL 寄存器　显示字符　DL 寄存器

17. 在 DOS 系统功能调用中，要实现字符串显示输出应使用_____号功能调用，要实现字符串输入应使用_____号功能调用。

答：9（09H）　10（0AH）

18. 在进行字符串输入的 DOS 系统功能调用时，如果键盘输入字符“ 345A ”后按回车键，则在键盘输入缓冲区的第二个单元的值是_____，在第三个到第七个单元的值依次是_____。

答：4　33H、34H、35H、41H、0DH

19. 在进行字符串显示输出的 DOS 系统功能调用时，要求输出字符串以_____结尾。

答：$（美圆符号）

20. 在进行 DOS 功能调用时，压堆栈的内容是_____。

答：PSW、CS、IP

7.4.4　简答题

1. 简述在程序设计中，采用子程序结构有哪些优点？

【解答】采用子程序结构的优点：

（1）简化了程序的设计过程，节省了程序的设计时间。
（2）缩短了程序的长度，节省了程序所占的存储空间。
（3）增加了程序的可读性，便于程序的修改和调试。
（4）便于程序的模块化、结构化和自顶向下的程序设计。

2. 简述过程的定义与子程序的定义有何区别？

【解答】过程的定义与子程序的定义区别在于：

（1）段的定义是以“名字 PROC 参数”开始，以“名字 ENDS”结束；而过程的定义是以“名字 PROC 参数”开始，以“名字 ENDP”结束。
（2）在参数的选择中，段的参数选择有定位方式、组合方式和类型；而过程的参数选择是 NEAR 和 FAR 两种类型。

3. 简述一个完整的子程序结构应包含哪几方面的内容？

【解答】一个完整的子程序结构应包含如下几方面的内容：

（1）为了便于程序具有一定的通用性，所以在子程序设计时要建立子程序的说明文件。
（2）在进行子程序设计时，其子程序是以过程的形式表示。在定义子程序时要说明它的类型，以便正确完成子程序的调用和返回。
（3）要注意现场的保护和现场的恢复。
（4）要正确地使用主程序和子程序之间的参数传递。
（5）在子程序中，除了要完成指定的功能外，还必须至少有一个出口（RET 指令）。

4. 在子程序的设计过程中，子程序的现场保护和现场恢复通常采用哪几种方法？哪一种方法使用较为方便？

【解答】在子程序的设计过程中，子程序的现场保护和现场恢复通常采用的方法有两种：第一种是利用压栈指令将寄存器的内容压堆栈实现现场的保护，利用弹栈指令将堆栈的内容送寄存器实现现场的恢复；第二种是利用数据传送指令将寄存器的内容送内存单元实现现场的保护，再利用数据传送指令将内存单元的内容送寄存器实现现场的的恢复。在程序的设计过程中，通常使用堆栈来保护和恢复现场较为方便。

5. 调用程序和子程序之间一般使用哪几种参数的传递方法？它们各自的特点是什么？

【解答】调用程序和子程序之间一般使用参数的传递方法有：寄存器法、约定单元法、堆栈法。它们各自的特点是：采用寄存器法传递参数主要是在 CPU 内部的各个寄存器间完成，它可以减少访问存储器的次数。但是 CPU 内部的寄存器是有限的，在有多个入口参数时这种方法就难以完成其操作。采用约定单元法传递参数可以将入口参数送指定单元，然后逐一完成入口参数的计算，并将计算的结果送指定单元。这种方法可以实现对多个参数的传递。采用堆栈法传递参数可以利用堆栈后进先出的操作原则实现参数的传递，这种方

法传递参数比约定单元法节省时间。

6. 简述段内调用和段间调用时堆栈有何不同？

【解答】在进行段内调用时，由于段地址不会变化，只是偏移地址发生变化，所以只是将偏移地址压堆栈，SP-2。在进行段间调用时，不但偏移地址发生变化，而且段地址也发生了变化，所以需要将段地址和偏移地址压堆栈，SP-4。

7. 简述子程序嵌套调用的基本原理。

【解答】如果在主程序（调用程序）中调用子程序时，在子程序的执行过程中又调用另外的一个子程序，这种子程序调用子程序的方法称为子程序的嵌套调用。

8. 简述递归调用的基本原理。

【解答】如果在主程序（调用程序）中调用子程序时，在子程序的执行过程中又直接或间接地调用该子程序，这种子程序调用称为子程序的递归调用。

9. 简述 DOS 系统功能调用的使用方法。

【解答】在进行 DOS 系统功能调用的方法是：

（1）将入口参数送相应的寄存器。
（2）将调用的功能号送 AH 寄存器。
（3）进行功能调用（即执行 INT　　N 指令）。
（4）出口参数分析。

10. 简述 DOS 系统功能调用和中断返回时堆栈的变化？

【解答】在进行 DOS 系统功能调用时，须将 PSW、CS 和 IP 的内容压堆栈，SP-6；在中断返回时，按照堆栈后进先出的操作原则，依次将调用时压堆栈的 PSW、CS 和 IP 的内容弹出送 IP、CS 和 PSW，SP+6，使 SP 指向 DOS 系统功能调用前的地址。

7.4.5 程序分析题

1. 现有程序如下：

```
STACK    SEGMENT     STACK  ‘STACK’
         DW      100 DUP (0)
STACK    ENDS
DATA     SEGMENT
BUF      DB      100 DUP (0)
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AX，STACK
```

```
          MOV     SS, AX
          LEA     DI, BUF
L:        MOV     AH, 1
          INT     21H
          CMP     AL, 0DH
          JZ      EXIT
          CALL    STO
          JMP     L
EXIT:     MOV     BYTE  PTR[DI],'$'
          LEA     DX, BUF
          MOV     AH, 9
          INT     21H
          MOV     AH, 4CH
          INT     21H
STO       PROC
          CMP     AL, 30H
          JB      NEXT
          CMP     AL, 39H
          JA      NEXT
          MOV     [DI], AL
          INC     DI
NEXT:     RET
STO       ENDP
CODE      ENDS
          END     START
```

请回答：（1）STO 子程序完成的功能是什么？

（2）该程序完成的功能是什么？

【解答】（1）STO 子程序的功能是：若 AL 寄存器中的内容为数字 ASCII 码，则将 AL 的内容送 DI 寄存器所指的单元。

（2）该程序的功能是将键盘输入的字符中的数字显示输出，其他字符则不显示，最后以回车结束。

2. **现有程序如下：**

```
STACK     SEGMENT     STACK  'STACK'
          DW      100 DUP (0)
STACK     ENDS
DATA      SEGMENT
BUF1      DB      100 DUP (?)
BUF2      DB      200, 0, 200 DUP (0)
EQ        DB      '两个字符串相等！$'
NEQ       DB      '两个字符串不相等！$'
FLAG      DB      0
DATA      ENDS
```

```
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     EX，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     DX，BUF2
        MOV     AH，0AH
        INT     21H
        MOV     CX，100
        LEA     SI，BUF1
        LEA     DI，BUF2+2
        CALL    SCMP
        CMP     FLAG，0
        JZ      N
        LEA     DX，EQ
        MOV     AH，9
        INT     21H
        JMP     EXIT
N：      LEA     DX，NEQ
        MOV     AH，9
        INT     21H
EXIT：   MOV     AH，4CH
        INT     21H
SCMP    PROC
        PUSH    AX
        MOV     AH，[DI－1]
        MOV     AL，9
        MOV     AH，0
        CMP     AX，CX
        JZ      R
        CLD
        CMPSB
        JNZ     R
        MOV     FLAG，0FFH
        JMP     R1
R：      MOV     FLAG，0
R1：     POP     AX
        RET
SCMP    ENDP
CODE    ENDS
        END     START
```

请回答：（1）SCMP **子程序完成的功能是什么？**

（2）该程序完成的功能是什么？

【解答】（1）SCMP 子程序的功能是比较两个字符串是否相等，若相等则将 0FFH 送 FLAG，若不相等则 FLAG 清 0。

（2）该程序的功能是从键盘输入字符串与 BUF1 字符串相比较，若相等则显示‘两个字符串相等！’，若不相等则显示‘两个不字符串相等！’。

3. 现有程序如下：

```
STACK    SEGMENT     STACK  ‘STACK’
         DW      100 DUP (0)
STACK    ENDS
DATA     SEGMENT
BUF      DB      20
C        DB      4
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AX，STACK
         MOV     SS，AX
         MOV     BL，C
         MOV     SI，BUF
L1：     PUSH    SI
         CALL    SUBP1
         CALL    SUBP2
         POP     SI
         INC     SI
         DEC     BL
         JNZ     L1
         MOV     AH，4CH
         INT     21H
SUBP1    PROC
         PUSH    AX
         PUSH    DX
L2：     MOV     DL，20H
         MOV     AH，2
         INT     21H
         DEC     SI
         JNZ     L2
         POP     DX
         POP     AX
         RET
SUBP1    ENDP
```

```
SUBP2    PROC
         PUSH    AX
         PUSH    DX
         MOV     CX，8
L3：     MOV     DL，‘*’
         MOV     AH，2
         INT     21H
         LOOP    L3
         MOV     DL，0DH
         MOV     AH，2
         INT     21H
         MOV     DL，0AH
         MOV     AH，2
         INT     21H
         POP     DX
         POP     AX
         RET
SUBP2    ENDP
CODE     ENDS
         END     START
```

请回答：（1）SUBP1 **子程序完成的功能是什么？**

（2）SUBP2 **子程序完成的功能是什么？**

（3）**该程序完成的功能是什么？**

【解答】（1）SUBP1 **子程序的功能是显示输出** SI **寄存器所表示的空格数。**

（2）SUBP2 **子程序的功能是显示输出** 8 **个字符‘*’。**

（3）**该程序的功能是在显示器上显示图形**

```
********
 ********
  ********
    ********
```

4. **现有程序如下：**

```
STACK    SEGMENT     STACK  ‘STACK’
         DW      100 DUP (0)
STACK    ENDS
DATA     SEGMENT
A        DB      30
B        DB      9
C        DW      5
DATA     ENDS
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA，SS：STACK
```

```
START：  MOV     AX, DATA
         MOV     DS, AX
         MOV     AX, STACK
         MOV     SS, AX
         MOV     CX, C
         MOV     BH, B
         MOV     BL, A
L1：     PUSH    BX
         CALL    SUBP1
         CALL    SUBP2
         POP     BX
         INC     BL
         SUB     BH, 2
         LOOP    L1
         MOV     AH, 4CH
         INT     21H
SUBP1    PROC
         PUSH    AX
         PUSH    DX
L2：     MOV     DL, 20H
         MOV     AH, 2
         INT     21H
         DEC     BL
         JNZ     L2
         POP     DX
         POP     AX
         RET
SUBP1    ENDP
SUBP2    PROC
         PUSH    AX
         PUSH    DX
L3：     MOV     DL,‘ * ’
         MOV     AH, 2
         INT     21H
         DEC     BH
         JNZ     L3
         MOV     DL, 0DH
         MOV     AH, 2
         INT     21H
         MOV     DL, 0AH
         MOV     AH, 2
         INT     21H
         POP     DX
         POP     AX
```

```
        RET
SUBP2   ENDP
CODE    ENDS
        END     START
```

请回答：（1）SUBP1 **子程序完成的功能是什么？**

（2）SUBP2 **子程序完成的功能是什么？**

（3）**该程序完成的功能是什么？**

【解答】（1）SUBP1 **子程序的功能是显示输出** BL **寄存器所表示的空格数。**

（2）SUBP2 **子程序的功能是显示输出字符‘*’，输出个数由** BH **寄存器所决定。**

（3）**该程序的功能是在显示器上显示图形**

```
*********
 *******
  *****
   ***
    *
```

5. **现有程序如下：**

```
STACK   SEGMENT     STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF1    DB      0E5H, 01H, 73H, 34H, 3AH, 0D5H
C1      =       $-BUF1
BUF2    DB      22H, 12H, 67H, 35H, 73H, 0B3H
C2      DB      $-BUF2
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     BX, BUF1
        MOV     CX, C1
        CALL    SORT
        LEA     BX, BUF2
        MOV     CX, C2
        CALL    SORT
        MOV     AH, 4CH
        INT     21H
```

```
SORT    PROC
        PUSH    AX
        PUSH    DX
        PUSH    SI
        PUSH    DI
        MOV     DX, CX
        DEC     DX
        MOV     SI, 1
L1:     MOV     DI, SI
        INC     DI
        MOV     AL, [BX+SI-1]
L2:     CMP     AL, [BX+DI-1]
        JBE     NEXT
        XCHG    [BX+DI-1], AL
        MOV     [BX+SI-1], AL
NEXT:   INC     DI
        CMP     DI, CX
        JBE     L2
        INC     SI
        CMP     SI, DX
        JBE     L1
        POP     DI
        POP     SI
        POP     DX
        POP     AX
        RET
SORT    ENDP
CODE    ENDS
        END     START
```

请回答：（1）**该子程序完成的功能是什么？**

（2）**程序执行完后，**BUF1 **和** BUF2 **的内容各是什么？**

【解答】（1）**子程序的功能是将以** BX **寄存器的内容为首地址的缓冲区中的** N **个无符号数按递增排序。**

（2）BUF1 **的内容为：**01H、34H、3AH、73H、0D5H、0E5H

BUF2 **的内容为：**12H、22H、35H、67H、73H、0B3H

6. **现有程序如下：**

```
STACK   SEGMENT     STACK  'STACK'
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      20E5H, 4501H, 7653H, 1234H, 354AH, 60D5H
C       =       ($-BUF) /2
```

```
SMAX    DW      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     AX，BUF
        PUSH    AX
        MOV     AX，C
        PUSH    AX
        CALL    MAX
        MOV     AH，4CH
        INT     21H
MAX     PROC
        PUSH    BP
        MOV     BP，SP
        PUSH    SI
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSHF
        MOV     SI，[BP＋6]
        MOV     CX，[BP＋4]
        MOV     BX，[SI]
        DEC     CX
        ADD     SI，2
        CLD
MAX1：   LODSW
        CMP     AX，BX
        JNA     NEXT
        XCHG    AX，BX
NEXT：   LOOP    MAX1
        MOV     SMAX，BX
        POPF
        POP     CX
        POP     BX
        POP     AX
        POP     SI
        POP     BP
        RET     4
MAX     ENDP
CODE    ENDS
```

```
        END     START
```

请回答：（1）该子程序完成的功能是什么？

（2）程序执行完后，SMAX 中的内容为何值？

【解答】（1）子程序的功能是将无符号字缓冲区 BUF 中的最大值送 SMAX 单元。

（2）程序执行后，SMAX 中的内容为 7653H。

7. **现有程序如下：**

```
STACK   SEGMENT STACK
        DB      100  DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      1064H
N       =       ($-BUF) /2
STR     DB      9  DUP (' ')
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, BUF
        MOV     CX, N
L:      MOV     AX, [SI]
        LEA     DI, STR
        CALL    CBD
        ADD     SI, 2
        LOOP    L
        MOV     AH, 4CH
        INT     21H
CBD     PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    DI
        PUSH    DI
        OR      AX, AX
        JNS     PLUS
        MOV     BYTE    PTR  [DI],'-'
        INC     DI
        NEG     AX
PLUS:   MOV     CX, 5
```

```
        MOV     BX，10
L1：    MOV     DX，0
        DIV     BX
        ADD     DL，30H
        MOV     [DI]，DL
        INC     DI
        OR      AX，AX
        JZ      L2
        LOOP    L1
L2：    MOV     BYTE    PTR[DI]，0DH
        INC     DI
        MOV     BYTE    PTR[DI]，0AH
        INC     DI
        MOV     BYTE    PTR[DI]，‘$’
        POP     DX
        MOV     AH，9
        INT     21H
        POP     DI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
CBD     ENDP
CODE    ENDS
        END     START
```

请回答：（1）该子程序完成的功能是什么？

（2）程序执行完后，显示输出的结果是什么？

【解答】（1）子程序的功能是将 BUF 缓冲区中的 16 位二进制数转换成十进制数并显示输出。

（2）程序执行后，显示输出的结果是 4196。

8. **现有程序如下：**

```
STACK   SEGMENT     STACK  ‘STACK’
        DW      100 DUP（0）
STACK   ENDS
DATA    SEGMENT
NUM1    DQ      7654321089ABCDEFH
NUM2    DQ      0FEDCBA9801234567H
RESUL   DT      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
```

```
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     ES，AX
         MOV     AX，STACK
         MOV     SS，AX
         LEA     SI，NUM1
         LEA     BX，NUM2
         LEA     DI，RESUL
         CALL    AD
         MOV     AH，4CH
         INT     21H
AD       PROC
         PUSH    AX
         PUSH    CX
         PUSHF
         CLC
         CLD
         MOV     CX，4
AGAIN：  LODSW
         ADC     AX，[BX]
         STOSW
         ADD     BX，2
         LOOP    AGAIN
         MOV     AX，0
         ADC     AX，0
         MOV     [DI]，AX
         POPF
         POP     CX
         POP     AX
         RET
AD       ENDP
CODE     ENDS
         END     START
```

请回答：（1）**该子程序完成的功能是什么？**

（2）**程序执行完后，**RESUL **单元的内容是什么？**

【解答】（1）**子程序的功能是将** NUM1 **和** NUM2 **的** 64 **位二进制数相加，结果送** RESUL **单元。**

（2）**程序执行后，**RESUL **单元的内容是** 75420EB90ABDF14601H

9. **现有程序如下：**

```
STACK    SEGMENT     STACK  ‘STACK’
         DW      100 DUP (0)
STACK    ENDS
```

```
DATA    SEGMENT
STR     DB      ‘PLEASE  INPUT  STRING (0…9)：$’
ASC     DB      20, 0, 20 DUP (0)
BCDB    DB      20 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     DX, STR
        MOV     AH, 9
        INT     21H
        LEA     DX, ASC
        MOV     AH, 10
        INT     21H
        LEA     BX, ASC+2
        MOV     CL, [BX-1]
        MOV     CH, 0
        LEA     SI, BCDB
        CALL    TAB
        MOV     AH, 4CH
        INT     21H
TAB     PROC
        PUSH    AX
        ADD     BX, CX
L：      DEC     BX
        MOV     AL, [BX]
        AND     AL, 0FH
        MOV     [SI], AL
        INC     SI
        LOOP    L
        POP     AX
        RET
TAB     ENDP
CODE    ENDS
        END     START
```

请回答：（1）该子程序完成的功能是什么？

（2）该程序完成的功能是什么？

【解答】（1）子程序的功能是将 BX 寄存器所指的 ASCII 码字符串转换成非压缩的 BCD 码并送 SI 寄存器所指的缓冲区保存。

（2）该程序完成的功能是将键盘输入的十进制数转换成非压缩的 BCD 码并

送 SI 寄存器所指的缓冲区保存。

10. 现有程序如下：

```
STACK   SEGMENT     STACK  'STACK'
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      8376H, 9028H, 4AB4H, 0947H, 3362H
N       =       ($-BUF) /2
STR     DB      7 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, BUF
        MOV     CX, N
LA:     MOV     AX, [SI]
        CALL    F2T10
        MOV     DL,'/'
        MOV     AH, 2
        INT     21H
        ADD     SI, 2
        LOOP    LA
        MOV     AH, 4CH
        INT     21H
F2T10   PROC
        PUSH    BX
        PUSH    DX
        PUSH    SI
        LEA     SI, STR
        OR      AX, AX
        JNS     PLUS
        NEG     AX
        MOV     [SI], BYTE  PTR'-'
        INC     SI
PLUS:   MOV     BX, 10
        CALL    BCP
        MOV     [SI], BYTE  PTR'$'
        LEA     DX, STR
        MOV     AH, 9
        INT     21H
```

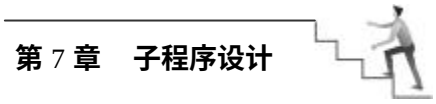
```
        POP     SI
        POP     DX
        POP     BX
        RET
F2T10   ENDP
BCP     PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    SI
        MOV     CX, 0
L1：    MOV     DX, 0
        DIV     BX
        PUSH    DX
        INC     CX
        OR      AX, AX
        JNZ     L1
L2：    POP     AX
        CMP     AL, 10
        JB      L3
        ADD     AL, 7
L3：    ADD     AL, 30H
        MOV     [SI], AL
        INC     SI
        LOOP    L2
        MOV     [SI+1], 0DH
        MOV     [SI+2], 0AH
        POP     SI
        POP     CX
        POP     BX
        POP     AX
        RET
BCP     ENDP
CODE    ENDS
        END     START
```

请回答：（1）F2T10 **子程序完成的功能是什么？**

（2）BCP **子程序完成的功能是什么？**

【解答】（1）F2T10 **子程序的功能是将** AX **寄存器中的带符号二进制数转换为十进制数并显示输出。**

（2）BCP **子程序的功能是将** AX **寄存器中的二进制数转换为十进制数的** ASCII **码并送** STR **字符串缓冲区保存。**



7.4.6　程序填空题

1. **下列程序是将** BUF1 **缓冲区的** 100 **个字单元的内容送** BUF2 **字缓冲区的子程序。请在程序的空格处填写适当的指令。**

```
B1TB2   PROC
        PUSH    AX
        PUSH    SI
                (1)
        PUSH    CX
        MOV     CX，100
        LEA     SI，BUF1
        LEA     DI，BUF2
L：     MOV     AX，[SI]
                (2)
                (3)
        POP     CX
        POP     DI
                (4)
        POP     AX
        RET
B1TB2   ENDP
```

【解答】（1）PUSH　DI
（2）MOV　[DI]，AX
（3）LOOP　L
（4）POP　SI

2. **下列程序是检查以** BX **寄存器的内容为首地址、以** CX **寄存器的内容为字符个数的字符串中是否有** AL **寄存器中存放的字符，如果有则将** FLAG **字节变量置** 1**；否则，清** 0 **的子程序。请在程序的空格处填写适当的指令。**

```
STRX    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DI
        CLD
                (1)
        SCASB
                (2)
        MOV     FLAG，1
                (3)
N：     MOV     FLAG，0
EXIT：  POP     DI
```

```
        POP     CX
        POP     BX
        POP     AX
              (4)
        STRX    ENDP
```

【解答】（1）MOV　DI，BX
（2）JNZ　N
（3）JMP　EXIT
（4）RET

3. **若** TAB **的内容为数组的首地址、**TAB＋2 **的内容为数组元数的个数、**TAB＋4 **的内容为数组和数的地址。下列程序是计算数组和数的子程序。请在程序的空格处填写适当的指令。**

```
ARYA    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DI
        MOV     BX，TAB
              (1)
        MOV     DI，TAB＋4
        MOV     DX，0
              (2)
L：     ADD     AX，[BX]
              (3)
        ADD     BX，2
        LOOP    L
              (4)
        MOV     [DI＋2]，DX
        POP     DI
        POP     CX
        POP     BX
        POP     AX
        RET
ARYA    ENDP
```

【解答】（1）MOV　CX，TAB＋2
（2）MOV　AX，0
（3）ADC　DX，0
（4）MOV　[DI]，AX

4. **下列程序是以** SI **寄存器的内容为压缩型** BCD **码的首地址、以** BX **寄存器的内容为字节的个数，将该压缩型** BCD **码显示输出的子程序。请在程序的空格处填写适当的指令。**

```
BCDP    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
L：     MOV     DL，[SI]
          (1)
        SHR     DL，CL
          (2)
        MOV     AH，2
        INT     21H
          (3)
        AND     DL，0FH
        ADD     DL，30H
        MOV     AH，2
        INT     21H
          (4)
        DEC     BX
        JNZ     L
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
BCDP    ENDP
```

【解答】（1）MOV　CL，4

（2）ADD　DL，30H　（OR　DL，30H）

（3）MOV　DL，[SI]

（4）INC　SI

5. **设以 SI 寄存器的内容为首地址、以 0 为结尾的无符号字数组。下列程序是从该数组中找出最大数的数值和数组的偏移地址送 DI 所指示的单元的子程序。请在程序的空格处填写适当的指令。**

```
MAXA    PROC
        PUSH    AX
        PUSH    SI
        PUSH    DI
          (1)
L：     CMP     [SI]，BYTE  PTR 0
        JZ      EXIT
        CMP     AX，[SI]
```

```
        (2)
        MOV     AX，[SI]
NEXT：  ADD     SI，2
        (3)
EXIT：  MOV     [DI]，AX
        (4)
        MOV     [DI＋2]，SI
        POP     DI
        POP     SI
        POP     AX
        RET
MAXA    ENDP
```

【解答】（1）MOV　AX，0　（XOR　AX，AX）
（2）JB　NEXT　（JBZ　NEXT **或** JNC　NEXT）
（3）JMP　L
（4）SUB　SI，2

6. **若** TAB、TAB＋2 **和** TAB＋4 **中的内容分别为压缩型** BCD **码的首地址，**TAB＋6 **中的内容是字节的个数。下列程序是将两个多位压缩型** BCD **码相加后送第三个缓冲区的子程序。请在程序的空格处填写适当的指令。**

```
BCDA    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    SI
        PUSH    DI
        MOV     SI，TAB
        MOV     BX，TAB＋2
        MOV     DI，TAB＋4
        MOV     CX，TAB＋6
        (1)
L：     MOV     AL，[SI]
        (2)
        DAA
        MOV     [DI]，AL
        INC     SI
        (3)
        INC     DI
        (4)
        POP     DI
        POP     SI
        POP     CX
        POP     BX
```

```
        POP     AX
        RET
BCDA    ENDP
```

【解答】（1）CLC

（2）ADC　　AL，[BX]

（3）INC　　BX

（4）LOOP　L

7. **下列程序是根据键盘输入的一位数据 N 求出 BUF 字节缓冲区中 N 个有符号数据的和数并送 DI 所指示的字单元的子程序。请在程序的空格处填写适当的指令。**

```
BUFA    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    DI
W：     MOV     AH，1
        INT     21H
        CMP     AL，30H
        JB      W
        CMP     AL，39H
        JA      W
                (1)
        MOV     CL，AL
                (2)
        LEA     BX，BUF
                (3)
        CLC
L：     MOV     AL，[BX]
                (4)
        ADD     DX，AX
        INC     BX
        LOOP    L
        POP     DI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
BUFA    ENDP
```

【解答】（1）AND　　AL，0FH

（2）MOV　　CH，0

（3）MOV　　DX，0

（4）CBW

8. **若 BX 寄存器中的内容为数组的首地址，CX 寄存器中的内容为数组元素的个数。下列程序是求数组元素的平均值和余数送 DI 寄存器所指示单元的子程序。请在程序的空格处填写适当的指令。**

```
ARYV    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    DI
        PUSH    CX
        MOV     AX，0
        MOV     DX，0
        ____(1)____
L：     ADD     AX，[BX]
        ____(2)____
        ADD     BX，2
        LOOP    L
        ____(3)____
        DIV     CX
        MOV     [DI]，AX
        ____(4)____
        POP     DI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
ARYV    ENDP
```

【解答】（1）CLC

（2）ADC　　DX，0

（3）POP　　CX

（4）MOV　　[DI＋2]，DX

9. **下列程序是将 AX 寄存器的 16 位无符号二进制数转换为十进制数并显示输出的子程序。请在程序的空格处填写适当的指令。**

```
DISP    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
```

```
        PUSH    DX
        MOV     CX, 0
        MOV     BX, 10
L:      ________(1)________
        CMP     AX, 0
        JZ      DISP0
        DIV     BX
        ________(2)________
        INC     CX
        JMP     L
DISP0:  CMP     CX, 0
        JZ      RE
        POP     DX
        ________(3)________
        MOV     AH, 2
        INT     21H
        DEC     CX
        ________(4)________
RE:     POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
DISP    ENDP
```

【解答】（1）MOV　DX，0

（2）PUSH　DX

（3）ADD　DL，30H

（4）JMP　DISP0

10. **下列程序是统计 BUF 字节缓冲区中 15 个单元中字母、数字和其他字符的个数并显示输出的子程序。请在程序的空格处填写适当的指令。**

```
DISPS   PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        MOV     CX, 15
        MOV     DX, 0
        MOV     BX, 0
        ________(1)________
L:      CMP     [BX], BYTE  PTR 30H
        JB      NNC
        CMP     [BX], BYTE  PTR 39H
```

```
        JBE     NUM
                (2)
        JB      NNC
        CMP     [BX], BYTE  PTR 5AH
        JBE     CHR
        CMP     [BX], BYTE  PTR 61H
        JB      NNC
        CMP     [BX], BYTE  PTR 7AH
                (3)
CHR：   INC     DL
        JMP     NEXT
NUM：   INC     DH
        JMP     NEXT
NNC：   INC     BL
NEXT：          (4)
        CALL    DIP
        MOV     DL, DH
        CALL    DIP
        MOV     DL, BL
        CALL    DIP
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
DISPS   ENDP
DIP     PROC
        PUSH    AX
        PUSH    DX
        MOV     AH, 2
        INT     21H
        MOV     DL, 0AH
        INT     21H
        MOV     DL, 0DH
        INT     21H
        POP     DX
        POP     AX
        RET
DIP     ENDP
```

【解答】（1）LEA　　BX，BUF（**或**　　MOV　　BX，OFFSET BUF）

（2）CMP　　[BX]，BYTE　PTR 41H

（3）JA　　NNC

（4）LOOP　　L

7.4.7　程序设计题

1. 请用寄存器法传递参数，编写一个将 BUF 缓冲区中的 16 位有符号数转换为 32 位补码的子程序。

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      8376H, 9028H, 4AB4H, 0947H, 3362H
N       =       ($-BUF) /2
BIN     DD      N  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, BUF
        LEA     DI, BIN
        MOV     CX, N
LA:     MOV     AX, [SI]
        CALL    F16T32
        ADD     SI, 2
        LOOP    LA
        MOV     AH, 4CH
        INT     21H
F16T32  PROC
        PUSH    DX
        CWD
        MOV     [DI], AX
        MOV     [DI+2], DX
        ADD     DI, 4
        POP     DX
        RET
F16T32  ENDP
CODE    ENDS
        END     START
```

2. 利用查表指令，编写一个将 X 中的压缩型 BCD 码转换为对应的 ASCII 码并依次存放在 BUF 缓冲区中的子程序。

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
```

```
STACK   ENDS
DATA    SEGMENT
X       DB      43H, 76H, 90H, 28H, 43H, 64H, 09H, 47H, 33H, 62H
N       =       $-X
BUF     DB      N+N  DUP (0)
ASCTB   DB      30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, X
        LEA     DI, BUF
        LEA     BX, ASCTB
        MOV     CX, N
LA：     MOV     AL, [SI]
        CALL    DTASC
        INC     SI
        LOOP    LA
        MOV     AH, 4CH
        INT     21H
DTASC   PROC
        PUSH    AX
        PUSH    CX
        MOV     AH, AL
        MOV     CL, 4
        SHR     AL, CL
        XLAT
        MOV     [DI], AL
        MOV     AL, AH
        AND     AL, 0FH
        XLAT
        MOV     [DI+1], AL
        ADD     DI, 2
        POP     CX
        POP     AX
        RET
DTASC   ENDP
CODE    ENDS
        END     START
```

3. **编写一个将 32 位二进制数转换为十进制数并显示输出的子程序。**

【解答】

```
STACK   SEGMENT STACK  'STACK'
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DD  32344376H, 2AD419028H, 342A4364H, 60050947H, 3362H
N       =       ($-BUF) /4
STR     DB      20  DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, BUF
        MOV     CX, N
LA:     LEA     DI, STR
        MOV     EAX, [SI]
        CALL    F2T10
        ADD     SI, 4
        LOOP    LA
        MOV     AH, 4CH
        INT     21H
F2T10   PROC
        PUSH    EAX
        PUSH    EBX
        PUSH    ECX
        PUSH    EDX
        PUSH    EDI
        MOV     CX, 0
        MOV     EBX, 10
L:      CMP     EAX, 0
        JZ      DISP
        MOV     EDX, 0
        DIV     EBX
        PUSH    DX
        INC     CX
        JMP     L
DISP:   CMP     CX, 0
        JZ      RE
        POP     AX
        ADD     AL, 30H
        MOV     [DI], AL
```

```
        INC     DI
        DEC     CX
        JMP     DISP
        MOV     [DI+2], BYTE PTR  '$'
        MOV     [DI+1], BYTE PTR  0DH
        MOV     [DI], BYTE PTR  0AH
RE:     LEA     DX, STR
        MOV     AH, 9
        INT     21H
        POP     EDI
        POP     EDX
        POP     ECX
        POP     EBX
        POP     EAX
        RET
F2T10   ENDP
CODE    ENDS
        END     START
```

4. **分别编写实现多位数的加法、减法、乘法的子程序。**

【解答】

```
STACK   SEGMENT STACK  'STACK'
        DW      100 DUP (0)
TOS     LABEL   WORD
STACK   ENDS
DATA    SEGMENT
A       LABEL   WORD
A1      DD      ?
A2      DD      ?
R       LABEL   WORD
AR      DW      0, 0
SR      DW      0, 0
MR      DW      0, 0, 0, 0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SP, TOS
        PUSH    A
        PUSH    A+2
        PUSH    A+4
        PUSH    A+6
```

```
        CALL    A32
        CALL    S32
        CALL    M32
        MOV     AH, 4CH
        INT     21H
A32     PROC
        PUSH    BP
        MOV     BP, SP
        PUSH    AX
        PUSH    BX
        MOV     AX, [BP+4]
        MOV     BX, [BP+8]
        ADD     AX, BX
        MOV     AR, AX
        MOV     AX, [BP+6]
        MOV     BX, [BP+0AH]
        ADC     AX, BX
        MOV     AR+2, AX
        POP     BX
        POP     AX
        POP     BP
        RET
A32     ENDP
S32     PROC
        PUSH    BP
        MOV     BP, SP
        PUSH    AX
        PUSH    BX
        MOV     AX, [BP+4]
        MOV     BX, [BP+8]
        SUB     AX, BX
        MOV     SR, AX
        MOV     AX, [BP+6]
        MOV     BX, [BP+0AH]
        SBB     AX, BX
        MOV     SR+2, AX
        POP     BX
        POP     AX
        POP     BP
        RET
S32     ENDP
M32     PROC
        PUSH    BP
        MOV     BP, SP
```

```
        PUSH    AX
        PUSH    BX
        PUSH    DX
        MOV     AX, [BP+4]
        MOV     BX, [BP+8]
        MUL     BX
        MOV     MR, AX
        MOV     MR+2, DX
        MOV     AX, [BP+4]
        MOV     BX, [BP+0AH]
        MUL     BX
        ADD     MR+2, AX
        ADC     MR+4, DX
        MOV     AX, [BP+6]
        MOV     BX, [BP+8]
        MUL     BX
        ADD     MR+2, AX
        ADC     MR+4, DX
        MOV     AX, [BP+6]
        MOV     BX, [BP+0AH]
        MUL     BX
        ADD     MR+4, AX
        ADC     MR+6, DX
        POP     DX
        POP     BX
        POP     AX
        POP     BP
        RET
M32     ENDP
CODE    ENDS
        END     START
```

5. **编写一个将 STR 字符串中的小写字母转换为大写字母，其他字符不变，并将转换后的字符串显示输出的子程序。**

【解答】

```
STACK   SEGMENT STACK  'STACK'
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
STR     DB      '3467RGBhfiDY6TzXTA686SfferrE'
N       =       $+STR
BUF     DB      N+1 DUP ('$')
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
```

```
START：  MOV     AX，DATA
         MOV     DS，AX
         MOV     AX，STACK
         MOV     SS，AX
         LEA     SI，STR
         LEA     DI，BUF
         MOV     CX，N
         CALL    ASC
         MOV     AH，4CH
         INT     21H
ASC      PROC
         PUSH    AX
         PUSH    DX
         PUSH    DI
L：      MOV     AL，[SI]
         CMP     AL，61H
         JB      NEXT
         CMP     AL，7AH
         JA      NEXT
         SUB     AL，20H
NEXT：   MOV     [DI]，AL
         INC     SI
         INC     DI
         LOOP    L
         MOV     [DI]，BYTE PTR‘ $ ’
         POP     DX
         MOV     AH，9
         INT     21H
         POP     DX
         POP     AX
         RET
ASC      ENDP
CODE     ENDS
         END     START
```

6. **编写一个从键盘输入字符串，然后显示该字符串中数字的个数、大写字母个数、小写字母个数和非数字字母个数的子程序。**

【解答】

```
STACK    SEGMENT STACK  ‘STACK’
         DW      100 DUP (0)
STACK    ENDS
DATA     SEGMENT
STR      DB      100，0，100 DUP (0)
NUM      DB      0
ACHR     DB      0
```

```
BCHR    DB      0
NCHR    DB      0
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     DX，STR
        MOV     AH，0AH
        INT     21H
        LEA     SI，STR
        CALL    TOJI
        MOV     AH，4CH
        INT     21H
TOJI    PROC
        PUSH    AX
        PUSH    CX
        MOV     CL，SI＋1
        MOV     CH，0
        ADD     SI，2
L：      MOV     AL，[SI]
        CMP     AL，30H
        JB      N3
        CMP     AL，39H
        JA      N1
        INC     NUM
        JMP     NEXT
N1：     CMP     AL，41H
        JB      N3
        CMP     AL，5AH
        JA      N2
        INC     ACHR
        JMP     NEXT
N2：     CMP     AL，61H
        JB      N3
        CMP     AL，7AH
        JA      N3
        INC     BCHR
        JMP     NEXT
N3：     INC     NCHR
NEXT：   INC     SI
        LOOP    L
```

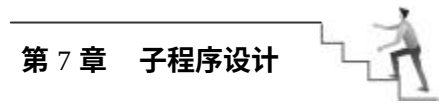
```
        MOV     AL, NUM
        CALL    DISP
        MOV     AL, ACHR
        CALL    DISP
        MOV     AL, BCHR
        CALL    DISP
        MOV     AL, NCHR
        CALL    DISP
        POP     CX
        POP     AX
        RET
TOJI    ENDP
DISP    PROC
        PUSH    AX
        PUSH    CX
        PUSH    DX
        MOV     CL, 4
        MOV     DL, AL
        SHR     DL, CL
        CMP     DL, 9
        JNA     D1
        ADD     DL, 7
D1：    ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     DL, AL
        AND     DL, 0FH
        CMP     DL, 9
        JNA     D2
        ADD     DL, 7
D2：    ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        MOV     DL,‘H’
        MOV     AH, 2
        INT     21H
        MOV     DL, 0AH
        MOV     AH, 2
        INT     21H
        MOV     DL, 0DH
        MOV     AH, 2
        INT     21H
        POP     DX
        POP     CX
```

```
        POP     AX
        RET
DISP    ENDP
CODE    ENDS
        END     START
```

7. **编写一个调用子程序的完整程序。其子程序的功能是将** BUF **缓冲区中的若干个** 16 **位有符号数转换为十进制数和八进制数并显示输出。**

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      3421H, -0A23H, 932DH, 4CF0H, 2457H
N       =       ($-BUF) /2
STR     DB      9 DUP (0)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     BX, BUF
        MOV     CX, N
L：      MOV     AX, [BX]
        LEA     SI, STR
        CALL    TDISP
        ADD     BX, 2
        LOOP    L
        MOV     AH, 4CH
        INT     21H
TDISP   PROC
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
        CALL    CRLF
        OR      AX, AX
        JNS     PLUS
        NEG     AX
        MOV     [SI], BYTE PTR‘-’
        INC     SI
PLUS：   MOV     BX, 10
        MOV     DX, 0
```



```
        MOV     CX, 0
LA:     OR      AX, AX
        JZ      RE
        DIV     BX
        PUSH    DX
        INC     CX
        JMP     LA
RE:     POP     DX
        ADD     DL, 30H
        MOV     [SI], DL
        INC     SI
        LOOP    RE
        MOV     [SI], BYTE PTR '$'
        MOV     AH, 9
        POP     DX
        INT     21H
        POP     DX
        POP     CX
        POP     BX
        RET
TDISP   ENDP
CRLF    PROC
        PUSH    AX
        PUSH    DX
        MOV     DL, 0AH
        MOV     AH, 2
        INT     21H
        MOV     DL, 0DH
        MOV     AH, 2
        INT     21H
        POP     DX
        POP     AX
        RET
CRLF    ENDP
CODE    ENDS
        END     START
```

8. **编写一个统计指定字缓冲区中的正数、负数和 0 的个数的子程序。要求在主程序中调用该子程序，将 BUF1 到 BUF3 中的数分别统计出来。**

【解答】

```
STACK   SEGMENT STACK  'STACK'
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF1    DW      3421H, -0A23H, 932DH, 4CF0H, 2457H
```

```
C1      DW      ($-BUF1)/2
P1      DB      0
N1      DB      0
Z1      DB      0
TAB1    DW      BUF1, C1, P1, N1, Z1
BUF2    DW      9000H, 6221H, -0023H, 0, 032DH, 40H, 27H
C2      DW      ($-BUF2)/2
P2      DB      0
N2      DB      0
Z2      DB      0
TAB2    DW      BUF2, C2, P2, N2, Z2
BUF3    DW      3001H, 0, 0, -03H, 0C000H, 6850H, 3532H
C3      DW      ($-BUF3)/2
P3      DB      0
N3      DB      0
Z3      DB      0
TAB3    DW      BUF3, C3, P3, N3, Z3
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA, SS:STACK
START:  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, TAB1
        CALL    PNZ
        LEA     SI, TAB2
        CALL    PNZ
        LEA     SI, TAB3
        CALL    PNZ
        MOV     AH, 4CH
        INT     21H
PNZ     PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    SI
        PUSH    DI
        PUSH    BP
        MOV     BX, [SI]
        MOV     DI, [SI+2]
        MOV     CX, [DI]
        MOV     BP, [SI+4]
        MOV     DI, [SI+6]
```

```
        MOV     SI, [SI+8]
LA：    MOV     AX, [BX]
        CMP     AX, 0
        JG      PLUS
        JL      NE
        INC     BYTE PTR [SI]
        JMP     NEXT
PLUS：  INC     BYTE PTR DS：[BP]
        JMP     NEXT
NE：    INC     BYTE PTR [DI]
NEXT：  ADD     BX, 2
        LOOP    LA
        POP     BP
        POP     DI
        POP     SI
        POP     CX
        POP     BX
        POP     AX
        RET
PNZ     ENDP
CODE    ENDS
        END     START
```

9. **编写一个搜索指定字缓冲区中的字符串是否有字符 N，若有则用字符 Y 取代字符 N 的子程序。在主程序中调用该子程序，将 BUF1 到 BUF3 中的字符 N 全部用 Y 替代。**

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF1    DB      ‘39WAGNTY9N9GSR8Tg7YogGGgf%ufR6’
C1      =       $-BUF1
BUF2    DB      ‘SRTHTH48658nTVTnKEDBNTLM5H’
C2      =       $-BUF2
BUF3    DB      ‘YYJY7U8KMNMUMNHY5667668nhbtTYYU’
C3      =       $-BUF3
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START：  MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, BUF1
        MOV     CX, C1
```

```
        CALL    NTY
        LEA     SI，BUF2
        MOV     CX，C2
        CALL    NTY
        LEA     SI，BUF3
        MOV     CX，C3
        CALL    NTY
        MOV     AH，4CH
        INT     21H
NTY     PROC
        PUSH    AX
        PUSH    CX
LA：    MOV     AL，[SI]
        CMP     AL，‘N’
        JNZ     CON
        MOV     [SI]，BYTE PTR‘Y’
        JMP     NEXT
CON：   CMP     AL，‘n’
        JNZ     NEXT
        MOV     [SI]，BYTE PTR ‘Y’
NEXT：  INC     SI
        LOOP    LA
        POP     CX
        POP     AX
        RET
NTYE    NDP
CODE    ENDS
        END     START
```

10. **设某班有 30 位同学，现需将某课程的成绩通过键盘输入并依次存放在 TAB 缓冲区中（得分范围 0～99）。采用子程序的结构形式编程，找出最高分并送显示器输出。要求：**

（1）编写一键盘输入子程序。

（2）编写一将两个数字的 ASCII 码转换成一个字节的压缩型 BCD 码的子程序。

（3）编写一将压缩型 BCD 码转换为 ASCII 码的子程序。

（4）编写一显示输出子程序。

（5）写出主程序调用子程序的程序段。

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
TAB     DB      30 DUP (0)
N       =       $-TAB
MAX     DB      0
```

```
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
        LEA     SI，TAB
        MOV     CX，N
        CALL    INDIS
        MOV     AH，4CH
        INT     21H
INDIS   PROC
        PUSH    AX
        PUSH    CX
        PUSH    SI
        PUSH    CX
        CALL    KBIN
        POP     CX
        POP     SI
        MOV     AL，0
LA：    CMP     AL，[SI]
        JNB     NEXT
        MOV     AL，[SI]
NEXT：  INC     SI
        LOOP    LA
        MOV     MAX，AL
        CALL    BBCD
        POP     CX
        POP     AX
        RET
INDIS   ENDP
KBIN    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
L1：    MOV     AH，1
        INT     21H
        CMP     AL，30H
        JB      L1
        CMP     AL，39H
        JA      L1
        MOV     BH，AL
L2：    MOV     AH，1
```

```
        INT     21H
        CMP     AL, 30H
        JB      L2
        CMP     AL, 39H
        JA      L2
        MOV     BL, AL
        CALL    ABCD
        MOV     [SI], BL
        INC     SI
        LOOP    L1
        POP     CX
        POP     BX
        POP     AX
        RET
KBIN    ENDP
ABCD    PROC
        PUSH    CX
        MOV     CL, 4
        SHL     BH, CL
        AND     BL, 0FH
        OR      BL, BH
        POP     CX
        RET
ABCD    ENDP
BBCD    PROC
        PUSH    AX
        PUSH    DX
        PUSH    CX
        MOV     CL, 4
        MOV     DL, AL
        SHR     DL, CL
        CALL    DISP
        AND     AL, 0FH
        MOV     DL, AL
        CALL    DISP
        POP     CX
        POP     DX
        POP     AX
        RET
BBCD    ENDP
DISP    PROC
        PUSH    AX
        MOV     AL, 2
        INT     21H
```

```
        POP     AX
        RET
DISP    ENDP
CODE    ENDS
        END     START
```

11. **设** TAB **中依次存放** 60 **位学生某一门课程的成绩，试编写一程序统计** 0～59、60～69、70～79、80～89、90～100 **的人数，并将统计结果以十进制数形式显示输出。**

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
TAB     DB      60 DUP (?)
N       =       $-TAB
STR     DB      0DH, 0AH,‘00～ 59 分的人数 ：00’; 一行字符占 20 字节
        DB      0DH, 0AH,‘60～ 69 分的人数 ：00’
        DB      0DH, 0AH,‘70～ 79 分的人数 ：00’
        DB      0DH, 0AH,‘80～ 89 分的人数 ：00’
        DB      0DH, 0AH,‘90～100 分的人数 ：00’
        DB      0DH, 0AH, 24H
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START： MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        LEA     SI, TAB
        LEA     DI, STR
        MOV     CX, N
        CALL    FENLEI
        MOV     AH, 4CH
        INT     21H
FENLEI  PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
L：     MOV     AL, [SI]
        CMP     AL, 60
        JAE     N1
        MOV     BX, 18
        CALL    TBCD
        JMP     NEXT
```

```
N1：    CMP     AL，70
        JAE     N2
        MOV     BX，38
        CALL    TBCD
        JMP     NEXT
N2：    CMP     AL，80
        JAE     N3
        MOV     BX，58
        CALL    TBCD
        JMP     NEXT
N3：    CMP     AL，90
        JAE     N4
        MOV     BX，78
        CALL    TBCD
        JMP     NEXT
N4：    MOV     BX，98
        CALL    TBCD
NEXT：  INC     SI
        LOOP    L
        MOV     DX，DI
        MOV     AH，9
        INT     21H
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
FENLEI  ENDP
TBCD    PROC
        PUSH    AX
        MOV     AX，[BX＋DI]
        XCHG    AH，AL
        INC     AL
        AAA
        OR      AX，3030H
        XCHG    AH，AL
        MOV     [BX＋DI]，AX
        POP     AX
        RET
TBCD    ENDP
CODE    ENDS
        END     START
```

12. **编写有子程序嵌套结构的程序，将 BUF 缓冲区中的若干个 16 位二进制数分别以二进制数、十进制数和十六进制数显示输出。**

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DW      75ABH, 0E95H, 9702H, 8321H, 1325H, 7AC0H
N       =       ($-BUF) /2
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE, DS：DATA, SS：STACK
START： MOV     AX, DATA
        MOV     DS, AX
        MOV     AX, STACK
        MOV     SS, AX
        MOV     CX, N
        LEA     BX, BUF
L：     MOV     AX, [BX]
        CALL    AMB
        ADD     BX, 2
        LOOP    L
        MOV     AH, 4CH
        INT     21H
AMB     PROC
        PUSH    AX
        PUSH    BX
        PUSH    SI
        MOV     SI, 2
        MOV     BH,‘B’
        CALL    TAN
        MOV     SI, 10
        MOV     BH,‘D’
        CALL    TAN
        MOV     SI, 16
        MOV     BH,‘H’
        CALL    TAN
        POP     SI
        POP     BX
        POP     AX
        RET
AMB     ENDP
TAN     PROC
        PUSH    AX
        PUSH    CX
        PUSH    DX
        PUSH    SI
```

```
        MOV     CX, 0
T0：    MOV     DX, 0
        DIV     SI
        PUSH    DX
        INC     CX
        OR      AX, AX
        JNZ     T0
T1：    POP     DX
        CMP     DL, 9
        JNA     T2
        ADD     DL, 7
T2：    ADD     DL, 30H
        MOV     AH, 2
        INT     21H
        LOOP    T1
        MOV     DL, BH
        MOV     AH, 2
        INT     21H
        MOV     DL, 0AH
        INT     21H
        MOV     DL, 0DH
        INT     21H
        POP     SI
        POP     DX
        POP     CX
        POP     AX
        RET
TAN     ENDP
CODE    ENDS
        END     START
```

13. **编写有子程序嵌套结构的程序，将键盘输入的字符串按数字、大写字母、小写字母和非数字字母分别显示输出。**

【解答】

```
STACK   SEGMENT STACK  ‘STACK’
        DW      100 DUP (0)
STACK   ENDS
DATA    SEGMENT
BUF     DB      100, 0, 100 DUP (0)
NUM     DB      100 DUP (0)
GCHR    DB      100 DUP (0)
LCHR    DB      100 DUP (0)
NCHR    DB      100 DUP (0)
STR     DB      ‘是否继续输入，是则键入字符Y！$’
DATA    ENDS
```

```
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA，SS：STACK
START： MOV     AX，DATA
        MOV     DS，AX
        MOV     AX，STACK
        MOV     SS，AX
L：     CALL    CRLF
        CALL    KBINP
        LEA     DX，STR
        MOV     AH，9
        INT     21H
        MOV     AH，1
        INT     21H
        CMP     AL，‘Y’
        JZ      L
        CMP     AL，‘y’
        JZ      L
        MOV     AH，4CH
        INT     21H
KBINP   PROC
        PUSH    AX
        PUSH    CX
        PUSH    SI
        LEA     DX，BUF
        MOV     AH，0AH
        INT     21H
        LEA     SI，BUF+2
        MOV     CL，[SI-1]
        MOV     CH，0
        LEA     BX，NUM
        LEA     DX，GCHR
        LEA     DI，LCHR
        LEA     BP，NCHR
        CALL    FEN
        LEA     DX，NUM
        MOV     AH，9
        INT     21H
        LEA     DX，GCHR
        MOV     AH，9
        INT     21H
        LEA     DX，LCHR
        MOV     AH，9
        INT     21H
        LEA     DX，NCHR
```

```
        MOV     AH, 9
        INT     21H
        POP     SI
        POP     CX
        POP     AX
        RET
KBINP   ENDP
FEN     PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
        PUSH    DI
        PUSH    BP
F1：    MOV     AL, [SI]
        CMP     AL, 30H
        JB      NC
        CMP     AL, 39H
        JNA     NU
        CMP     AL, 41H
        JB      NC
        CMP     AL, 5AH
        JNA     GC
        CMP     AL, 61H
        JB      NC
        CMP     AL, 7AH
        JNA     LC
NC：    MOV     DS：[BP], AL
        INC     BP
        MOV     DS：[BP], BYTE PTR ‘ $ ’
        JMP     NEXT
NU：    MOV     [BX], AL
        INC     BX
        MOV     [BX], BYTE PTR ‘ $ ’
        JMP     NEXT
LC：    MOV     [DI], AL
        INC     DI
        MOV     [DI], BYTE PTR ‘ $ ’
        JMP     NEXT
GC：    PUSH    BX
        MOV     BX, DX
        MOV     [BX], AL
        INC     BX
```

```
        MOV     [BX], BYTE PTR '$'
        MOV     DX, BX
        POP     BX
NEXT:   INC     SI
        LOOP    F1
        POP     BP
        POP     DI
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
FEN     ENDP
CRLF    PROC
        PUSH    AX
        PUSH    DX
        PUSH    SI
        MOV     AH, 2
        MOV     DL, 0AH
        INT     21H
        MOV     DL, 0DH
        INT     21H
        POP     DX
        POP     AX
        RET
CRLF    ENDP
CODE    ENDS
        END     START
```