

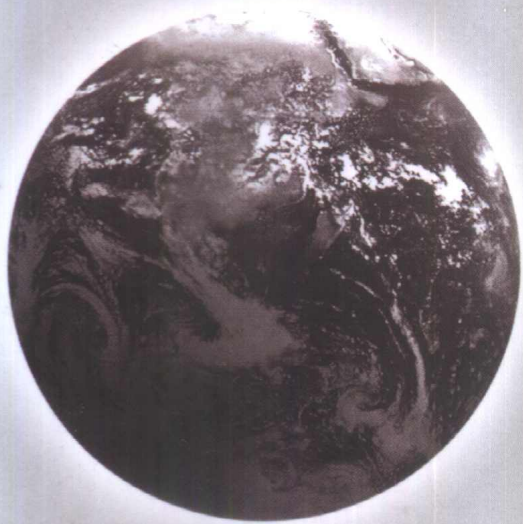
全国高等教育自学考试指定教材配套辅导丛书

高级语言程序设计

应试指导及模拟试题

▼ 全国高等教育自学考试命题研究组 编
▼ 教材依据 迟成文 主编

J I S U A N J I Z H U A N Y E



计算机类 权威辅导

- ▶ 重点难点精讲
- ▶ 解题技巧分析
- ▶ 教材同步训练
- ▶ 考前实战演习

中国大地出版社

397

全国高等教育自学考试指定教材辅导

高级语言程序设计

应试指导及模拟试题

全国高等教育自学考试命题研究组 编

教材依据 迟成文 主编

中国大地出版社

图书在版编目 (CIP) 数据

高级语言程序设计应试指导及模拟试题 / 全国高等教育自学考试命题研究组编. —北京: 中国大地出版社, 2002.3

(全国高等教育自学考试辅导丛书)

ISBN 7-80097-498-7

I. 高… II. 全… III. 高级语言—程序设计—高等教育—自学考试—自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 012504 号

责任编辑: 王慧军

出版发行: 中国大地出版社

社址邮编: 北京市海淀区大柳树路 19 号 100081

电 话: (010) - 62183493 (发行部)

传 真: (010) - 62183493

印 刷: 北京市顺义康华福利印刷厂

开 本: 787 × 1092 1/16

印 张: 220

字 数: 4000 千字

版 次: 2002 年 3 月第 1 版

印 次: 2002 年 3 月第 1 次印刷

印 数: 3600 册

书 号: ISBN 7 - 80097 - 498 - 7/TP·7

定 价: 300.00 元 (全套)

(凡购买中国大地出版社的图书, 如发现印装质量问题, 本社发行部负责调换)

前 言

国家教育部考试中心于 2002 年开始,正式执行自学考试新计划,同时使用新编的大纲和教材。

参加自考的学生渴求在考前能通过应试指导的帮助及模拟试题的演练,全面检查自己所学的知识是否扎实,考试大纲所要求的内容是否掌握,已经理解的知识能否完整、确切、简明地进行书面表述,并借此增强考生分析和解决实际问题的能力,帮助考生顺利通过考试。因此,为配合广大考生参加考试,并能顺利过关,我们利用多年积累的自考教学辅导资源和经验,全面系统地剖析了各门专业课程新大纲和教材的内容体系,组织编写了一套“全国高等教育自学考试应试指导及模拟试题”丛书,推向全国,以满足考生之急需,适应社会之需要。

本书在编写过程中,严格按照考试大纲的要求,以指定教材为基础,包括了所有考试的知识点,并着重突出重点和难点,充分体现了“在考察课程主体知识的同时,注重考查能力尤其是应用能力”的新的命题指导思想。

本书以习题为主,完全按照指定教材的结构,以章为单位。每章设“考试要求”、“知识重点”、“反馈测试题解”三部分。“考试要求”主要是考试大纲所规定的本章考核要求。“知识重点”主要是对该章的重点、要点内容的总结归纳;“反馈测试题解”则根据考试大纲对各知识点不同能力层次的要求,将知识及知识点下的细目以各种主要考试题型的形式编写,覆盖全部考核内容,适当突出重点章节,并且加大重点内容的覆盖密度,所有试题均附详细解答;书后附有模拟试卷及 2001 年度试题,供考生检验自己学习情况,建议在规定时间内完成。本书由张燕利主编。

欢迎广大读者对本丛书提出宝贵意见,以便我们今后工作中得以改进。

全国高等教育自学考试命题研究组

2002.3

目 录

高级语言程序设计考试概述	(1)
第一章 概 述	(11)
◎考试要求	(11)
◎知识重点	(11)
◎反馈测试题解	(13)
第二章 基本数据类型和运算	(19)
◎考试要求	(19)
◎知识重点	(20)
◎反馈测试题解	(22)
第三章 顺序结构、选择结构和循环结构的程序设计	(33)
◎考试要求	(33)
◎知识重点	(34)
◎反馈测试题解	(40)
第四章 数组	(75)
◎考试要求	(75)
◎知识重点	(75)
◎反馈测试题解	(78)
第五章 指针	(99)
◎考试要求	(99)
◎知识重点	(99)
◎反馈测试题解	(103)
第六章 函数	(140)
◎考试要求	(140)
◎知识重点	(140)
◎反馈测试题解	(143)
第七章 结构型、共用型和枚举型	(185)
◎考试要求	(185)
◎知识重点	(185)
◎反馈测试题解	(192)
第八章 文件	(228)
◎考试要求	(228)
◎知识重点	(228)
◎反馈测试题解	(235)

第九章 编译预处理与带参数的主函数	(261)
◎考试要求	(261)
◎知识重点	(261)
◎反馈测试题解	(264)
高级语言程序设计考前模拟试题(一)	(293)
高级语言程序设计考前模拟试题(一)参考答案	(298)
高级语言程序设计考前模拟试题(二)	(301)
高级语言程序设计考前模拟试题(二)参考答案	(308)
高级语言程序设计考前模拟试题(三)	(311)
高级语言程序设计考前模拟试题(三)参考答案	(322)
高级语言程序设计考前模拟试题(四)	(326)
高级语言程序设计考前模拟试题(四)参考答案	(337)
高级语言程序设计考前模拟试题(五)	(340)
高级语言程序设计考前模拟试题(五)参考答案	(345)
二〇〇一年下半年全国高等教育自学考试高级语言程序设计试卷及参考答案	(347)

高级语言程序设计考试概述

高级语言程序设计是高等教育自学考试计算机及应用专业(专科)中一门重要的专业基础课,它既可以为其它专业课程奠定程序设计的基础,又可以作为其它专业课程的程序设计工具。

设置本课程的主要目的是通过对 C 语言的语法规则、数据类型、数据运算、语句、系统函数、程序结构的学习,掌握应用 C 语言进行简单程序设计的技能,为进行各种实用程序开发奠定一个良好的基础。

通过本课程的学习,应能达到知识和技能两方面的目标。

1. 知识方面:

(1)熟练掌握 C 语言的数据类型(基本类型、构造类型、指针类型等)和各类运算符,能正确使用表达式实现各种数据的简单加工。

(2)熟练掌握 C 程序的三种基本结构(顺序、选择、循环)的特点,能使用相关语句完成这三种基本结构的程序设计任务。

(3)掌握 C 语言的常用库函数使用,以及用户函数的定义、调用、参数传递等方法。

2. 技能方面:

(1)熟练掌握阅读和分析简短程序的方法和技巧。

(2)熟练掌握设计和调试简短程序的方法及技巧。

(3)了解并初步掌握实用程序的开发与调试技术。

考试方式为闭卷、笔试。考试时间为 150 分钟。评卷采用 100 分制,60 分为及格。考试时,只允许携带钢笔和铅笔,答卷规定用钢笔完成。各章的考核要求中所列出的所有细目都是考试的内容。试题覆盖到章,命题要突出重点章节,要加大重点内容的覆盖密度。本课程对不同能力层次要求在试卷中所占比例大致如下:“识记”占 20%，“领会”占 30%，“简单应用”占 30%，“综合应用”占 20%。试卷中的难易程度比例大约为易:较易:较难:难 = 2:3:3:2。试卷题型有:单项选择题、填充题、程序分析题、程序设计题。

(一)单项选择题

例 1. 当执行下面程序且输入“ABC”时,输出的结果是

()

```
#include <stdio.h>
#include <string.h>
main() {
    char ss[10] = "12345"; strcat(ss, "6789");
    gets(ss); printf("%s \n", ss);
}
```

A. ABC

B. ABC9

C. 123456ABC

D. ABC456789

分析:函数 `strcat()` 是字符串连接。`gets()` 中输入数据时以回车符作为结束符。在存入内存时,输入字符串末尾的回车符自动就换成代码 `'\0'`,从而使其成为字符串。它在 `stdio.h` 中被定义,输出 `ss` 时,遇

到“\0”停止输出。可知本题应选择 A。

答:A

例 2. 合法的数组定义是

()

A. int a[] = "tring";

B. int a[5] = {0, 1, 2, 3, 4, 5};

C. char a = "string";

D. char a[] = {0 1 2 3 4 5};

分析:A 错,C 语言规定不能把一个字符常量赋给一个整型数组;

B 错,这是因为实际赋值的元素数超过定义数组的元素个数;

C 错,不能用赋值语句将字符常量赋给一个字符;

D 正确,在 C 语言中没有定数组长度而对数组全部元素赋值,系统会据此定义数组 a 的长度为 6。

答:D

例 3. 以下程序的输出结果是

()

A. -1 1

B. 0 1

C. 1 0

D. 0 0

```
main()
{
    int a = -1, b = 1, k;
    if( ++a < 0) &&! (b-- <= 0)
        printf("%d %d\n", a, b);
    else
        printf("%d %d\n", b, a);
}
```

分析:本题程序中,(++a < 0)的值为 0,所以不再作(b-- <= 0)的运算,可知 a = 0, b = 1,从而执行 printf("%d %d\n", b, a)的显示结果是:1 0。

答:C

例 4. 请读下列程序:

```
#include <stdio.h>
main()
{
    int x = 1, y = 0, a = 0, b = 0;
    switch(x)
    {
        case 1:
            switch(y)
            {
                case 0:
                    a++; break;
                case 1:
                    b++; break;
            }
        case 2:
            a++; b++; break;
    }
    printf("a = %d, b = %d\n", a, b);
}
```

上面程序的输出结构是

()

A. a = 2, b = 1

B. a = 1, b = 1

C. a = 1, b = 0

D. a = 2, b = 2

分析:本题为双重开关分支语句,外层开关语句当 x = 1 时转入内层开关执行,由于 y = 0,所以执行

语句“a++”后 a=1。所以,最后结果为 a=1,b=0。本题应选择 A。

答:A

例 5.若 k 是 int 型变量,且下面的程序片段:

```
k = -3;
if(k <= 0) printf("###")
else printf("&&&");
```

上面程序片段的输出结果是

()

A. ###

B. &&&

C. ### &&&

D. 有语法错误,无输出结果

分析:if 语句后的 printf 语句末尾缺少“;”,因此有语法错误。本题应选择 D

注意:分号“;”是 C 语言中语句的组成部分,不能缺少,否则不能构成语句。

答:D

例 6.以下程序的输出结果是

()

```
#include <stdio.h>
main()
{printf("%d\n", NULL);}
```

A. 不确定(因变量无定义)

B. 0

C. -1

D. 1

分析:本题中的 NULL 是 C 语言中的预定义标识符,代表字符‘\0’,即代表 ASCII 码为 0 的字符,所以当 %d 格式输出时,应输出对应字符的 ASCII 码,也即是输出 0。

答:B

例 7.设有语句“int a=3;”,则执行了语句“a+=a-=a*a;”后,变量 a 的值是

()

A. 3

B. 0

C. 9

D. -12

分析:C 语言可以包含复合赋值语句,赋值语句是按从右到左进行结合的,本题中“a++=a-=a*a”的执行步骤为:

(1)运算“a-=a*a”,这时有 $a = a - a * a = 3 - 3 * 3 = -6$;

(2)运算“a+=-6”,这时有 $a = a + 6 = 0$ 。

答:B

例 8.以下程序的输出结果是

()

```
#include <stdio.h>
main()
{int a=2,c=5;printf("a=%%d,b=%%d\n",a,c);}
```

A. a= %2,b= %5

B. a=2,b=5

C. a= %%d, %%d

D. a= %d,b= %d

分析:在 C 语言的 printf 函数语句中,如在格式说明中有两个连续的 %% 字符,则 %% 不再作为格式描述字符使用,而是当成一个字符“%”的原样输出;对于格式符少于输出项时,多余的输出项不予输出。在本题中,两处 %% 都作为字符 % 处理,所以,两项 a 与 c 都无输出格式说明符,从而不予输出。本题应选择 D。

答:D

例 9.以下程序的输出结果是

()

```
main()
{int a=12,b=12;printf("%d %d\n",--a,++b);}
```

A. 10 10

B. 12 12

C. 11 10

D. 11 13

分析: $-a$ 与 $++b$ 为前置运算, 首先进行复合赋值运算, 可知 a 的值为 11, b 的值为 13, 然后再进行输出。所以, 本题应选择 D。

答: D

例 10. 语句 `printf("%d", (a=2)&&(b=-2));` 的输出结果是 ()

A. 无输出

B. 结果不确定

C. -1

D. 1

分析: 由于 $(a=2)$ 的值为 2, 而 $(b=-2)$ 的值为 -2, 所以可知 $(a=2)&&(b=-2)$ 的值为 1。

答: D

(二) 填空题

例 1. 阅读下列程序:

```
#include <stdio.h>
main() {
    int i, j, row, column, m; static int array[3][3] = {{100, 200, 300}, {28, 72, -30}, {-850, 2, 6}};
    m = array[0][0];
    for(i = 1; i < 3; i++)
        for(j = 0; j < 3; j++)
            if(array[i][j] < m) {m = array[i][j]; row = i; column = j;}
    printf("%d, %d, %d\n", m, row, column);
}
```

上述程序的输出结果是_____。

分析: 本题程序中定义了一个整型数组 `array`, 并为其赋初值。数组 `array` 的值如下所示:

$$\begin{bmatrix} 100 & 200 & 300 \\ 28 & 72 & -30 \\ -850 & 2 & 6 \end{bmatrix}$$

程序中使用两层嵌套的 `for` 循环语句及一个 `if` 语句从数组的第一行开始将 `array` 中每个元素同变量 `m` 进行比较, 使变量 `m` 中保持存放较小的元素的值, 变量 `row` 存放较小元素的行下标, 变量 `column` 存放较小元素的列下标。整个嵌套的 `for` 循环执行完后, `array` 中全部元素都同 `m` 进行一遍比较, 此时 `m` 中的值为 `array` 中最小元素的值, 变量 `row` 与 `column` 分别代表此最小元素的行下标和列下标。程序中最后一条 `printf` 函数调用语句用于分别输出 `m`、`row`、`column` 3 个变量的值。所以, 应为: -850, 2, 0。

答: -850, 2, 0

例 2. 下面的 `findmax` 函数返回数组 `s` 中最大值元素的下标, 数组中元素的个数由 `t` 传入, 请将程序补充完整。

```
findmax(int s[], int t) {
    int k, p;
    for(p = 0, k = p; p < t; p++) if(s[p] > s[k]) _____
    return k;
}
```

分析: 函数 `findmax` 用于返回数组中最大值元素的下标, 从 `return` 语句可发现, 变量 `k` 表示最大值元素的下标, 求最大值的下标由 `for` 循环语句实现。算法要点是: 先假定第一个元素最大, 并记下其位置

k = p, 然后依次将数组中的后续元素 s[p] 与此假设的最大元素 s[k] 进行比较, 当出现某个元素的值大于假设的最大值时 (s[p] > s[k]), 记下此较大值的下标。可知此处应为 "k = p;"。

注意: C 语言中除复合语句而外, 必须以分号结束。

答: k = p

例 3. 下面 invert 函数的功能是将一个字符串 str 的内容颠倒过来。请将程序补充完整。

```
#include <stdio.h>
void invert(char str[]) {
    int i, j, (1) ;
    for(i = 0, j = strlen(str) (2) ; i < j; i++, j--) { k = str[i]; str[i] = str[j]; str[j] = k; }
```

分析: 在 C 语言中, 所有变量都必须在使用之前进行说明, 所以可以看出 (1) 处应为 K。由 for 循环语句的一般形式知道, (2) 处应完成对循环变量 j 赋初值, 根据题意, 要把串 str 中内容颠倒, i 的初值应取 0, j 的初值应是串 str 中要作颠倒的元素个数 (不包含串结尾标志 '\0'), 由于数组的下标从零开始, 从而最后一个字符即第 strlen(str) 个字符的下标应为 strlen(str) - 1, 所以, j 的初值应为 strlen(str) - 1, 即 (2) 处应为 - 1。

答: (1) k (2) - 1

例 4. 以下程序用来对从键盘上输入的两个字符串进行比较, 然后输出两个字符串中的第一个不相同字符的 ASCII 码之差, 例如: 输入的两个字符串分别为 abcdefg 和 abceef, 则输出为 - 1, 请填空。

```
#include <stdio.h>
main() {
    char str1[100], str2[100], c; int i, s;
    printf(" \nInput string1: \n"); gets(str1);
    printf(" \nInput string2: \n"); gets(str2);
    i = 0;
    while((str1[i] == str2[i] && (str1[i] != (1)))) i++;
    s = (2); printf("%d \n", s);
}
```

分析: 本题程序中比较两个字符串的方法是使用 while 语句把两个 PB 字符串中相应字符依次取出, 当遇到两个字符不相等或第一个字符串结束时终止循环, 然后再输出不相等的字符串之差 (如两字符串相等, 则同时结束, 这时两个字符都是 '\n', 其差为 0)。从而可知 (1) 处应为 '\n' 或 0, (2) 处应为 str1[i] - str2[i]。

答: (1) '\n' 或 0 (2) str1[i] - str2[i]

例 5. 以下程序可把从键盘上输入的十进制数 (long 型) 以二到十六进制数的形式输出, 请填空。

```
#include <stdio.h>
main() {
    char b[16] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'}; int c[64], d, i = 0, base; long n;
    printf("Enter a number: \n"); scanf("%ld", &d); printf("Enter new base: \n"); scanf("%d", &base);
    do { c[i] = (1); i++; n = n/base; } while(n != 0);
    printf("Transmute new base: \n");
    for(--i; i >= 0; --i) { d = c[i]; printf("%c", b (2)); }
```

分析:本题程序中把一个十进制数 n 转化为 $base$ 进制数,采用“除 $base$ 取余法”,即从低位开始以 n 除以 $base$ 的余数作为转化后的一个位数,再以 n 除以 $base$ 的商作为新的 n ,继续进行转化。从而可知,(1)处应为 n 除以 $base$ 的余数,即 $n \% base$, (2)处用于显示所转化的 $base$ 进制数,把 $b[d]$ 输出,故(2)处应填 $[d]$ 。

答:(1) $n \% base$ (2) $[d]$

例 6. 以下两条 if 语句:

```
if(a <= b) x = 1;
else y = 2
if(a > b) printf(" * * * y = %d \ n", y);
else printf(" # # # x = %d \ n", x);
```

可合并成一条 if 语句为_____。

分析:首先将两条 if 语句写为如下等价语句:

```
if(a <= b) x = 1;
else y = 2;
if(a <= b) printf(" # # # x = %d \ n", x);
else printf(" * * * y = %d \ n", y);
```

从而可合并为:

```
if(a <= b) { x = 1; printf(" # # # x = %d \ n", x); }
else { y = 2; printf(" * * * y = %d \ n", y); }
```

答: $\text{if}(a <= b) \{ x = 1; \text{printf}(" \# \# \# x = \%d \setminus n", x); \}$

$\text{else } \{ y = 2; \text{printf}(" * * * y = \%d \setminus n", y); \}$

例 7. 下面程序的输出结果是_____。

```
# include < stdio. h >
main() {
    int a = 3, b = 2, c = 1;
    c = + + b;
    b * = a + c
    { int b = 5, c = 12; c / = b * 2; a - = c; printf("%d, %d, %d, ", a, b, c); a + = - - c; }
    printf("%d, %d, %d, ", a, b, c);
}
```

分析:本题程序中,函数 main 的首部定义了 3 个整型变量 a 、 b 、 c ,按照它们的初值,可计算执行复合语句之前 c 的值为 -2 , b 的值为 3 。在进入复合语句后,变量 b 、 c 被重新定义,并赋初值,用户可计算出在复合语句中, a 、 b 、 c 的值分别是 2 、 5 、 1 ,所以第一个 printf 语句将输出: $2, 5, 1$;最后执行复合语句中最后一条语句可知, a 的值为 2 。

由于变量 b 和 c 在 main 首部及复合语句中均被说明,在复合语句中,说明 b 、 c 的作用域只限制在本复合语句,所以执行复合语句之后的 printf 语句的 b 、 c 的输出值与复合语句中的 b 、 c 值无关,而应取复合语句之前的两个赋值语句的计算结果,也就是 b 的值为 3 , c 为 -2 ,可知第 2 个 printf 语句的输出为 $2, 3, -2$;用户可知该处应为: $2, 5, 1, 2, 3, -2$ 。

答: $2, 5, 1, 2, 3, -2$

(三) 程序分析

例 1. 下面程序的运行结果为

()

```

main()
{
    int i = 5;
    printf("%d \ n", sub(i));
}

sub(n) int n;
{
    int a;
    if(n == 1) return 1;
    a = n + sub(n - 1);
    return(a);
}

```

答:15

例 2. 下面程序的运行结果为

()

```

#include < stdlib.h >
main()
{
    int * p1, * p2[5], k;
    if((p1 = malloc(100 * sizeof(int))) != NULL)
    {
        p2[0] = p1;
        for(k = 1; k < 5; k++)
            p2[k] = p2[k - 1] + 7;
        for(k = 0; k < 100; k++)
            * p1++ = k;
        for(k = 0; k < 5; k++)
            printf("%3d", * p2[k]);
    }

    printf("\ n");
}

```

答:0,7,14,21,28

例 3. 下面程序的运行结果为()。

```

main()
{
    char ss[10] = "a program";
    int num = 4;
    sub(ss, 0, num - 1); sub(ss, num, 6); sub(ss, 0, 6);
    printf("%s \ n", ss);
}

sub(ss, num1, num2)
char * ss; int num1, num2;
{
    char t, * p;
    p = ss + num2; ss = ss + num1;
    while(ss < p) { t = * ss; * ss = * p; * p = t; ss++; p--; }
}

```

```
}
```

答: ogra pram

例 4. 若运行下列程序时, 输入以下指定数据, 请回答下面问题, 将答案填入对应的() 中。

```
main()
{
    int w,s,sum();
    w = 1;
    while(w! = 0)
    {
        scanf("%d",&w);
        if(w! = 0)
        {
            s = sum(w);
            printf("%d,%d \ n",w,s);
        }
    }
}

int sum(n)
int n;
{
    int s;
    static int total = 0;
    if (n <= 50) s = 1 * n;
    else      s = 50 * 1 + (n - 50) * 2;
    total + = s;
    return( total );
}
```

输入数据(第一列): 10 20 0<回车>

问题:

①此程序的运行结果为(1)

②若将 sum 函数中的语句:

```
static int total = 0;
```

改为:

```
int total = 0;
```

则程序的运行行为(2)。

答: (1) 10, 10 (2) 20, 30

例 5. 下面程序的运行结果为()。

```
#include <stdio.h>
int abc(int u,int v);
void main()
{ int a = 24, b = 16, c;
  c = abc(a,b);
```

```

        printf("value = %d",c);
    }
int abc(int u,int v)
{ int w;
  while(v)
  { w = u%v;
    u = v;
    v = w;
  }
  return u;
}

```

答: value = 8

例 6. 下面程序的运行结果为()。

```

#include <stdio.h>
main()
{ int i = 4, j;
  for(j = 0; j < i; j++)
    pic(28 - j, "");
    pic(i + 2 * j, ' * ');
    putchar('\n');
  for(j = 2; j >= 0; j--)
  {
    pic(30 - j, "");
    pic(i + 2 * j, ' * ');
    putchar('\n');
  }
}
pic(len, c)
int len;
char c;
{
  int k;
  for(k = 1; k <= len; k++) putchar(c);
}

```

答:

```

* * * * *
* * * * *
* * * * *
* * * *

```

(四)程序设计题

例 1. 编一程序, 将两个字符串连接起来, 不要用 strcat 函数。

答:main()

```
{
    char s1[80],s2[40];
    int i=0,j=0;
    printf("\n input string1:");
    scanf("%s",s1);
    printf("\n input string2:");
    scanf("%s",s2);
    while(s1[i]!='\0')
        i++;
    while(s2[j]!='\0')
        s1[i++]=s2[j++];
    s1[i]='\0';
    printf("\n new string: %s",s1);
}
```

例 2. 编写一个程序,将字符数组 S2 中的全部字符拷贝到字符数组 S1 中。不用 strcpy 函数。拷贝时,' \0'也要拷贝过去。' \0'后面的字符不拷贝。

答:

```
#include "stdio.h"
main()
{
    char from[80],to[80];
    int i;
    printf("input string:");
    scanf("%s",from);
    for(i=0;i<=strlen(from);i++)
        to[i]=from[i];
    printf("copied string: %s \n",to);
}
```

第一章 概 述

通过本章学习,要求了解程序、程序设计、高级语言的概念,认识 C 语言的字符集、词类和语句的分类;了解 C 程序的基本结构,为后续章节的学习奠定一个基础。

重点:C 语言的字符集和保留字;C 语言词类和语句的分类;C 程序的基本结构。

◎考试要求

1. 程序、程序设计和高级语言,达到“识记”层次。
程序的概念。
程序设计的任务和主要步骤。
高级语言的概念与作用。
2. C 语言和 C 程序,达到“识记”层次。
C 语言的发展。
C 程序的基本结构。
3. C 语言的字符集、词类和句类,达到“领会”层次。
C 语言的字符集。
C 语言的词的分类原则。
C 语言的保留字及其基本作用。
C 语言的保留字及其基本作用。
C 语言的语句分类。

◎知识重点

(一)程序是对解决某个问题的方法步骤的描述

从计算机角度来说,程序是某种计算机能理解并执行的计算机语言描述解决问题的方法步骤。它按照一定的顺序一步一步执行。而程序设计就是分析解决问题的方法步骤。

语言是人机对话中的一个不可缺的工具。最早的语言是二进制语言,使用计算机能直接识别和执行的二进制代码来编写程序,后来又出现了用符号代替二进制码的汇编语言,它所编写的程序一般只能在同类型的计算机上运行,所以又称为“面向机器的语言”。

C 语言是一种可以在任何计算机上运行的计算机语言,是一种高级语言。它提供的语句可以描述复杂的加工处理过程,是在 B 语言的基础上发展起来的,其前身是 ALGOL 语言(第一个高级语言,第二个高级语言是 FORTRAN)。

高级语言必须翻译成二进制码才能执行,编译有两种方式:一种是翻译一句执行一