

全国计算机等级考试



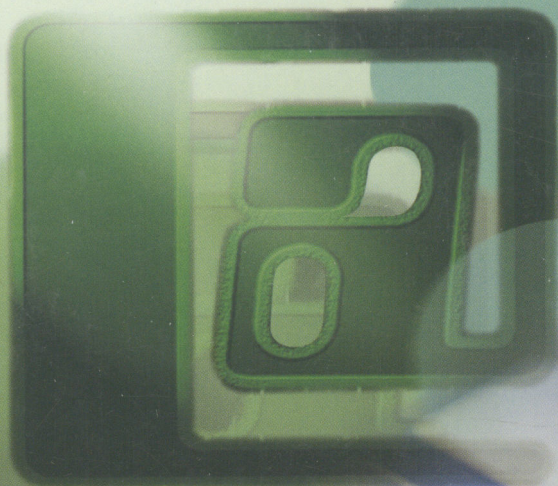
历届上机真题详解

二级 C 语言程序设计

(2008版)

全国计算机等级考试命题研究组 编

南开大学出版社



全国计算机等级考试

历届上机真题详解

二级 C 语言程序设计

(2008 版)

全国计算机等级考试命题研究组 编

南开大学出版社

天 津

图书在版编目(CIP)数据

全国计算机等级考试历届上机真题详解: 2008 版. 二级 C 语言程序设计 / 全国计算机等级考试命题研究组编. 5 版. —天津: 南开大学出版社, 2007. 11

ISBN 978-7-310-02282-3

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

中国版本图书馆 CIP 数据核字(2007)第 154591 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

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

*

天津泰宇印务有限公司印刷

全国各地新华书店经销

*

2007 年 11 月第 5 版 2007 年 11 月第 5 次印刷

787×1092 毫米 16 开本 29.75 印张 750 千字

定价: 42.00 元

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

前言

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

为了帮助考生更顺利地通过计算机等级考试, 我们做了大量市场调查, 根据考生的备考体会, 以及培训教师的授课经验, 推出了《历届上机真题详解——二级 C 语言程序设计》。本书主要由如下部分组成。

一、二级 C 语言上机考试题库和解析

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

本题库中, 填空题、改错题和编程题各 100 套, 每道题都配有清晰准确的答案以及详尽的分析, 精解考点, 分析题眼, 详解重点难点, 剖析程序流程, 给出应试技巧。另外, 许多题目都有一个“提高补遗”部分, 说明了如何更巧妙地解题, 可以开阔读者的编程思路, 这可以说是本书的画龙点睛之笔。本书题量巨大, 题目典型, 是备考二级 C 语言程序设计上机考试的极佳参考。

二、上机考试必备知识

在这部分中, 讲解上机考试的注意事项以及上机考试过程, 并针对 C 语言上机考试的改错题和编程题, 分别提出了如何准备和应对的方法与技巧。

三、机试全真模拟环境配套光盘

本书配套光盘中有: (1) 全真模拟上机环境, 您可以在这个环境中进行考前实战训练, 检验自己的实际水平, 本书所有习题均可在系统中练习答题和评分; (2) 考试过程的录像动画演示, 从登录、答题到交卷, 均有指导教师的全程语音讲解。

本书针对参加全国计算机等级考试二级 C 语言程序设计的考生, 同时也可以作为普通高校、大专院校、成人高等教育以及相关培训班的练习题和考试题使用。

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

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

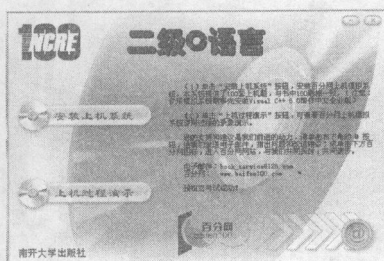
电子邮件: book_service@126.com

百分网: www.baifen100.com

全国计算机等级考试命题研究组

2007 年 10 月

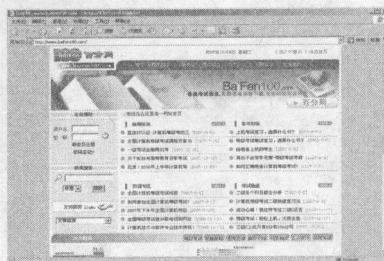
配套光盘说明



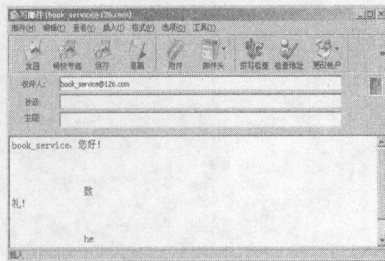
光盘初始启动界面，可选择安装上机系统、查看上机操作过程



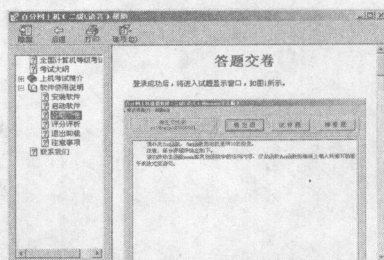
上机操作过程的录像演示，有指导教师的全程语音讲解



单击光盘初始界面的百分网图标，您可以在百分网，您可以在此与我们共同探讨问题



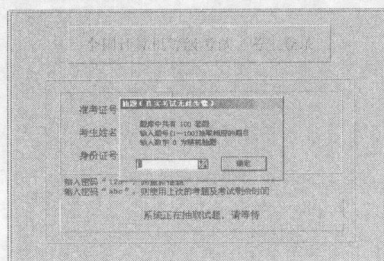
单击光盘初始界面左下角的邮件图标，您可以给我们发送邮件，提出您的建议和意见



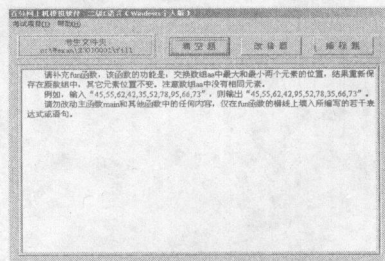
从“开始”菜单可启动帮助系统，在这里可看到考试简介、考试大纲以及详细的软件使用说明



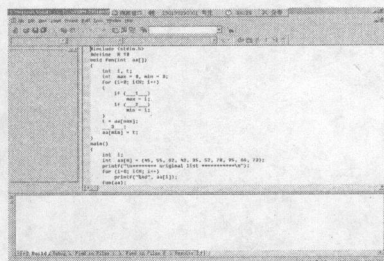
双击桌面上的软件名称启动上机系统，按照提示登录



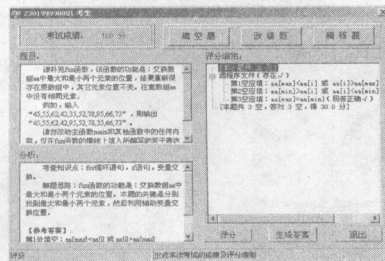
您可以随机抽题，也可以指定固定的题目



浏览题目界面，查看考试题目，单击“考试项目”开始答题



在实际环境答题，完成后单击工具栏中的“交卷”按钮



答案和分析界面，查看所考核题目的答案和分析

目 录

第一部分 程序填空题	1
第二部分 改错题	146
第三部分 编程题	279
第四部分 上机考试必备知识	455
4.1 复习经验谈	455
4.2 考场指南	456
4.2.1 上机考题类型	456
4.2.2 熟悉考试环境	456
4.2.3 等待评分结果	456
4.2.4 考试注意事项	456
4.2.5 改错题答题策略与技巧	458
4.2.6 编程题答题策略与技巧	459
4.3 上机考试过程	462
4.3.1 登录	462
4.3.2 答题	463
4.3.3 交卷	466

第一部分 程序填空题

★★

第1题

请补充 fun 函数，fun 函数的功能是求 n 的阶乘。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序：

```
#include <stdio.h>
long fun(int n)
{
    if (____1____)
        return (n*fun(____2____));
    return ____3____;
}
main()
{
    printf("10!=%ld\n", fun(10));
}
```

【答案】

第1处填空：n>1 或 1<n

第2处填空：n-1

第3处填空：1 或 1l 或 1L

【解析】

考查知识点：采用递归算法求数的阶乘。

解题思路：所谓递归法，可以这样来理解：若要求 10!，则可看作是 9!×10，因此要求 10!，则需要知道 9!；同理，9!=8!×9，则要求知道 8!；以此类推，直到知道了 1! 的值，才能得到 10! 的值。这一过程就是递归问题的第一个阶段——回推。而 1! 值（通常初始值都是已知的或能很容易求得的）是已知的（1!=1）。这里需要注意，必须要有条件使递归调用终止，否则就会成为无终止调用，成为死循环。则通过已知条件 1!=1，推得下一个结果 2!=1!×2=1×2=2。同理，也可推得 3!、4! 直到推得 10!。这一过程就是递归问题的第二个阶段——递推。递推的共同特点是从一个已知的事实出发，按一个共同的规律推出下一个事实，再从这个新的已知事实出发，再向下推出一个新的事实。

因此可用下面的递归公式表示 n!

$$n! = \begin{cases} 1 & (n \leq 1) \\ n \times (n-1)! & (n > 1) \end{cases}$$

【答案详解】

```
long fun(int n)
{
    if (n > 1)                /*第1空: 首先进行判断, 如果 n>1 则返回 n*(n-1)!,
                                否则返回 1*/
        return (n*fun(n-1)); /*第2空: 此处应返回 n*(n-1)!, 即 n*fun(n-1)*/
    return 1;                 /*第3空: 执行到这里 n 肯定小于等于 1, 故应返回 1*/
}
```

★★

第2题

请在函数 fun 的横线上填写若干表达式, 使从键盘上输入一个整数 n, 输出斐波纳契数列。斐波纳契数列是一种整数数列, 其中每数等于前面两数之和, 如: 0 1 1 2 3 5 8 13

注意: 部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容, 仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序:

```
#include <stdio.h>
int fun(int n);
main()
{
    int i, n = 0;
    scanf("%d", &n);
    for (i=0; i<n; i++)
        printf("%d ", fun(i));
}
int fun(int n)
{
    if (____1____)
        return 0;
    else if (____2____)
        return 1;
    else
        return ____3____;
}
```

【答案】

第1处填空: $n==0$ 或 $0==n$ 或 $!n$

第2处填空: $n==1$ 或 $1==n$

第3处填空: $fun(n-1)+fun(n-2)$ 或 $fun(n-2)+fun(n-1)$

【解析】

考查知识点: 采用递归算法求斐波纳契数列。

解题思路: 递归算法见第1题。采用递归算法必须能够从已知条件求得值, 根据斐波纳契数列算法, 后面的数都是其前面两个数的和, 因此求后面的数必须回归到求它前面的两个数, 以此回推, 直到已知条件。

【答案详解】

```

int fun(int n)
{
    if (n==0)                /*第1空:第1个已知条件,当n==0时,返回0*/
        return 0;
    else if (n==1)           /*第2空:第2个已知条件,当n==1时,返回1*/
        return 1;
    else
        return fun(n-1)+fun(n-2); /*第3空:第n项等于第n-1和第n-2项之和*/
}

```

★★

第3题

请补充函数 fun(char *s), 该函数的功能是把字符串中的内容逆置。

例如: 字符串中原有的字符串为 abcde, 则调用该函数后, 串中的内容为 edcba。

注意: 部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容, 仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序:

```

#include <string.h>
#include <conio.h>
#include <stdio.h>
#define N 81
void fun(char *s)
{
    int i, n = strlen(s)-1;
    char t;
    for (i=0; i<n; i++, ___1___)
    {
        t = s[i];
        ___2___;
        ___3___;
    }
}
main()
{
    char a[N];
    printf("Enter a string:");
    gets(a);
    printf("The original string is:");
    puts(a);
    fun(a);
    printf("\n");
    printf("The string after modified:");
    puts(a);
}

```

【答案】

第 1 处填空: $n--$ 或 $--n$ 或 $n-1$ 或 $n=n-1$

第 2 处填空: $s[i]=s[n]$ 或 $*(s+i)=*(s+n)$

第 3 处填空: $s[n]=t$ 或 $*(s+n)=t$

【解析】

考查知识点: 字符串, 循环语句。

解题思路: fun 函数的功能是把字符串中的内容逆置。本题的思路是: (1) 定义一个字符变量 t 辅助字符交换; (2) 利用双循环变量分别从两头开始遍历字符串, 直到中间相遇, 把字符串后面的字符与其相对应的前面字符进行互换。此题的关键在于如何表达前后对应的下标, 字符串首字符的下标为 0, 字符串尾字符的下标为字符串长度-1, 每当前面的下标增 1 时, 后面与它对应的下标也应减 1。

【答案详解】

```
void fun(char *s)
{
    int i, n = strlen(s)-1;
    char t;
    for (i=0; i<n; i++, n--) /*第 1 空: 当前面下标 i 增 1 时, 后面下标 n 应减 1*/
    {
        t = s[i];
        s[i] = s[n];          /*第 2 空: 交换算法第 2 步,
                               把后面的字符拷贝到前面与它对应的位置*/
        s[n] = t;             /*第 3 空: 交换算法第 3 步,
                               把 t 所保存的前面对应字符拷贝到后面对应的位置*/
    }
}
```

★★

第 4 题

请补充函数 fun, 它的功能是: 计算并输出 n (包括 n) 以内能被 3 或 7 整除的所有自然数的倒数之和。

例如, 在主函数中从键盘给 n 输入 30 后, 输出为: $s=1.226323$ 。

注意: 部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容, 仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序:

```
#include <stdio.h>
double fun(int n)
{
    int i;
    double sum = 0.0;
    for (i=1; ___1___; i++)
        if (i%3==0 ___2___ i%7==0)
            sum += ___3___/i;
    return sum;
}
```

```

}
main()
{
    int n;
    double s;
    printf("\nInput n: ");
    scanf("%d", &n);
    s = fun(n);
    printf("\n\ns=%f\n", s);
}

```

【答案】

第1处填空: $i \leq n$ 或 $n \geq i$

第2处填空: `||`

第3处填空: 1.0 或 (double)1

【解析】

考查知识点: 循环语句, 条件表达式, 数据类型强制转换。

解题思路: fun 函数的功能是计算并输出 n (包括 n) 以内能被 3 或 7 整除的所有自然数的倒数之和。这里需要注意: (1) 筛选范围是 $1 \sim n$ (包括 n), 所以循环变量 i 初始为 1, 在 $\leq n$ 范围内循环; (2) 判断是否能被 3 或 7 整除, 条件或应该用 “`||`” 运算符; (3) 自然数的倒数是小数, 而 i 为整数, 因此分子位置的 1 应为与 `sum` 相同的 `double` 型, 可以用实型常量 1.0, 也可以用强制类型转换 `(double)1`。

【答案详解】

```

double fun(int n)
{
    int i;
    double sum = 0.0;
    for (i=1; i<=n; i++)          /*第1空: 筛选范围应包括n, 因此 i<=n*/
        if (i%3==0 || i%7==0)    /*第2空: 条件或应该用“||”运算符*/
            sum += 1.0/i;        /*第3空: 为使“/”运算的结果为double类型,
                                   此处可填1.0或(double)1*/
    return sum;
}

```

★★

第5题

请补充函数 fun, 其功能是: 计算并输出给定 10 个数的方差:

$$S = \sqrt{\frac{1}{10} \sum_{k=1}^{10} (X_k - X')^2}$$

$$\text{其中 } X' = \frac{1}{10} \sum_{k=1}^{10} X_k$$

例如, 给定的 10 个数为 15.0, 19.0, 16.0, 15.0, 18.0, 12.0, 15.0, 11.0, 10.0, 16.0, 输出为

s=2.758623。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序：

```
#include <stdio.h>
#include <math.h>
double fun(double x[10])
{
    int i;
    double avg=0.0;
    double sum=0.0;
    double abs=0.0;
    double sd;
    for (i=0;i<10;i++)
        sum += ____1____;
    avg=sum/10;
    for (i=0;i<10;i++)
        ____2____ (x[i]-avg)*(x[i]-avg);
    sd = ____3____ (abs/10);
    return sd;
}
main()
{
    double s, x[10]={15.0,19.0,16.0,15.0,18.0,12.0,15.0,11.0,10.0,16.0};
    int i;
    printf("\nThe original data is :\n");
    for(i=0;i<10;i++)printf("%6.1f",x[i]);
    printf("\n\n");
    s=fun(x);
    printf("s=%f\n\n",s);
}
```

【答案】

第 1 处填空：x[i] 或 *(x+i)

第 2 处填空：abs += 或 abs = abs +

第 3 处填空：sqrt

【解析】

考查知识点：循环语句，数学函数。

解题思路：fun 函数的功能是计算并输出给定 10 个数的方差。本题需要了解方差的算法，即使不清楚，也可以从题目给出的表达式写出来。本函数要求调用<math.h>函数库中的函数 sqrt() 求平方根。

【答案详解】

```
double fun(double x[10])
{
    int i;
```

```

double avg=0.0;
double sum=0.0;
double abs=0.0;
double sd;
for (i=0;i<10;i++)
    sum+=x[i];                /*第1空: 累计数组 x 中 10 个元素的和*/
avg=sum/10;
for (i=0;i<10;i++)
    abs+=(x[i]-avg)*(x[i]-avg); /*第2空: 累加数组 x 中
                                10 个元素与平均值 avg 的差的平方*/
sd=sqrt(abs/10) ;            /*第3空: 调用数学函数 sqrt 开方以求方差*/
return sd;
}

```

★★

第6题

下列给定程序中，函数 fun 的功能是：对 N 名学生的学习成绩，按从高到低的顺序找出前 m ($m \leq 10$) 名学生来，并将这些学生数据存放在一个动态分配的连续存储区中，此存储区的首地址作为函数值返回。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序：

```

#include <stdio.h>
#include <stdlib.h>
#define N 10
typedef struct ss
{
    char num[10];
    int order;
} STU;
STU *fun(STU a[], int m)
{
    STU b[N], *tt;
    int i,j,k;
    tt = __1__(sizeof(STU)*m);
    for(i=0; i<N; i++)
        b[i]=a[i];
    for(k=0; k<m; k++)
    {
        for(i=j=0; i<N; i++)
            if(b[i].order __2__ b[j].order)
                j=i;
        tt[k]=b[j];
        b[j].order=0;
    }
    return __3__;
}

```

```

    }
    outresult(STU a[], FILE *pf)
    {
        int i;
        for(i=0; i<N; i++)
            fprintf(pf, "No=%s Mark=%d\n", a[i].num, a[i].order);
        fprintf(pf, "\n\n");
    }
    main()
    {
        STU a[N]={ {"A01", 80}, {"A02", 79}, {"A03", 66}, {"A04", 82}, {"A05", 87},
                    {"A06", 93}, {"A07", 78}, {"A08", 60}, {"A09", 85}, {"A10", 73}};
        STU *p_order;
        int i, m;
        printf("***** The Original data *****\n");
        outresult(a, stdout);
        printf("\nGive the number of the students who have better score: ");
        scanf("%d", &m);
        while(m>10)
        {
            printf("\nGinve the number of the students who have better score:");
            scanf("%d", &m);
        }
        p_order=fun(a,m);
        printf("***** THE RESULT *****\n");
        printf("*****The top students*****\n");
        for(i=0; i<m; i++)
            printf("    %s    %d\n", p_order[i].num, p_order[i].order);
        free(p_order);
    }

```

【答案】

第 1 处填空: (STU*)malloc

第 2 处填空: >

第 3 处填空: tt

【解析】

考查知识点: 结构体, 动态分配内存, 函数返回。

解题思路: 函数 fun 的功能对 N 名学生的学习成绩, 按从高到低的顺序找出前 m ($m \leq 10$) 名学生来, 并将这些学生数据存放在一个动态分配的连续存储区中, 此存储区的首地址作为函数值返回。fun 函数用到了动态分配内存, 因此要注意 malloc() 函数的调用格式。

【答案详解】

```

STU *fun(STU a[], int m)
{
    STU b[N], *tt;
    int i, j, k;
    tt=(STU*)malloc(sizeof(STU)*m);    /*第 1 空: 给 tt 分配一个动态的连续存储区,
                                           所以应该调用函数 malloc(), 并将其返回值
                                           强制转换为 STU 的指针类型*/

```

```

for(i=0; i<N; i++)
    b[i]=a[i];
for(k=0; k<m; k++)
{
    for(i=j=0; i<N; i++)
        if(b[i].order>b[j].order) /*第2空: 从下面的语句 j=i 来看, j 被用来
                                    指向最高的成绩, 所以只有当 b[i].order>
                                    b[j].order 时, 才需要将 i 赋给 j*/

        j=i;
    tt[k]=b[j];
    b[j].order=0;
}
return tt; /*第3空: 返回动态存储区首地址 tt*/
}

```

★★

第7题

请补充函数 fun, 该函数的功能是: 删去一维数组中所有相同的数, 使之只剩一个。数组中的数已按由小到大的顺序排列, 函数返回删除后数组中数据的个数。

例如, 若一维数组中的数据是: 1 1 1 2 2 2 3 4 4 5 5 6 6 7 7 8 9 9 10 10。

删除后, 数组中的内容应该是: 1 2 3 4 5 6 7 8 9 10。

注意: 部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容, 仅在 fun 函数的横线上填入所编写的若干表达式或语句。

试题程序:

```

#include <stdio.h>
#define N 80
int fun(int a[], int n)
{
    int i,t,j=0;
    t=a[0];
    for(i=1;i<n;i++)
        if(____1____)
            ;
        else
        {
            ____2____;
            t=a[i];
        }
    a[j++]=t;
    return j;
}
main()
{
    int a[N]={1,1,2,2,2,3,4,4,5,5,6,6,6,7,7,8,9,9,10,10}, i,n=20;
    printf("The original data :\n");
    for(i=0; i<n; i++)

```

```

        printf("%4d",a[i]);
    n=fun(a,n);
    printf("\n\nThe data after deleted :\n");
    for(i=0;i<n;i++)
        printf("%4d",a[i]);
    printf("\n ");
}

```

【答案】

第 1 处填空: $t==a[i]$ 或 $a[i]==t$

第 2 处填空: $a[j++] = t$

【解析】

考查知识点: 数组操作。

解题思路: fun 函数的功能是删去一维数组中所有相同的数,使之只剩一个。本题的关键是如何删除相同的数,其算法是:先用一个临时变量 t 记录数组中出现的元素,如果后面的元素相同则考查其后面的数,如果没有相同的,则把 t 赋给新数组,把数组出现的非相同元素赋给 t 继续比较。

【答案详解】

```

int fun(int a[], int n)
{
    int i,t,j=0;
    t=a[0];           /*把 a 数组的第一个元素赋给 t 作为初始值*/
    for(i=1;i<n;i++)  /*遍历数组的所有元素*/
        if(t==a[i])  /*第 1 空: 如果第 i 个元素与 t 相同,
                        则执行空语句,继续比较下一个元素*/
            ;
        else
        {
            a[j++] = t; /*第 2 空: 如果不相同则把 t 赋给原数组保存*/
            t=a[i];     /*把下一个数组元素赋给 t*/
        }
    a[j++] = t;        /*把最后一个数保存在数组 a 中*/
    return j;          /*返回数组元素的个数*/
}

```

★★

第 8 题

请补充 main 函数,该函数的功能是:从键盘输入一组整数,使用条件表达式找出最大的整数。当输入的整数为 0 时结束。

例如,输入 1 2 3 5 4 0 时,最大的数为 5。

注意:部分源程序给出如下。

仅在横线上填入所编写的若干表达式或语句,勿改动函数中的其他任何内容。

试题程序:

```

#include <stdio.h>
#include <conio.h>

```

```

#define N 100
main()
{
    int num[N];
    int i = -1;
    int max = 0;
    printf("\nInput integer number: \n");
    do
    {
        i++;
        printf("num[%d]=", i);
        scanf("%d", __1__);
        max = __2__ num[i] : max;
    } while(__3__);
    printf("max=%d\n", max);
}

```

【答案】

第1处填空: &num[i] 或 num+i

第2处填空: max<num[i]? 或 num[i]>max?

第3处填空: num[i]!=0 或 0!=num[i]

【解析】

考查知识点: 标准输入语句, 条件表达式, do...while 循环。

解题思路: fun 函数的功能是利用条件表达式找出从键盘输入的一组整数的最大值。条件表达式的语法格式为: 表达式1? 表达式2: 表达式3, 如果表达式1成立, 则整个表达式的值为表达式2的值, 否则为表达式3的值。do...while 循环语句至少要运行一次, 当 while 后面的条件不成立时其循环终止。

【答案详解】

```

main()
{
    int num[N];
    int i=-1; /*循环变量, 初值为-1*/
    int max=0;
    printf("\nInput integer number: \n");
    do
    {
        i++; /* 循环变量自增 1*/
        printf("num[%d]=", i);
        scanf("%d", &num[i]); /*第1空: 标准输入函数 scanf 要求使用地址值, 因此应填 num[i] 的地址*/
        max=max<num[i]?num[i]:max; /*第2空: 根据后面表达式 2 和表达式 3 的变量, 可以确定前面应用小于号*/
    }while(num[i]!=0); /*第3空: 循环终止条件*/
    printf("max=%d\n", max);
}

```

★★