

教育部考试中心指定教材配套辅导
National Computer Rank Examination

全国计算机 等级考试

NCR
Examination
全国计算机等级考试

上机考试习题集 三级 C 语言程序设计

(适用于三级网络、数据库、信息管理技术)

考试研究中心 主编

光盘 **附** 参考答案
可调试程序

中科多媒体电子出版社

全国计算机等级考试试题集

三级 C 语言程序设计

考试研究中心 主编

中科多媒体电子出版社

内 容 提 要

本系列根据教育部考试中心 2002 年公布的《全国计算机等级考试大纲》的要求编写。本册的特点是：知识点全且概括精炼，例题分析透彻，提供了大量与真实考试难度及题型相当的考题供考生训练，使读者顺利通过考试。

未经许可，不得以任何方式复制或抄袭光盘及配书之部分或全部内容。
版权所有，翻版必究。

书 名：全国计算机等级考试试题集—— 三级 C 语言程序设计

电脑制作：北京暴风雪科技有限公司

出 版：中科多媒体电子出版社

印 刷：北京市富生印刷厂

经 销：各地新华书店

开 本：787×1092 1/16 印张：12

版 次：2003 年 8 月第 1 版

凡购买的图书和光盘有问题者，请向购买处调换。

电话：(010) 62522622、13801352164

致读者

经过几年的全国计算机等级考试辅导实践，我们得到了全国广大考生的极大支持。在这几年的过程中，大量的考生纷纷来信、来电或发 e-mail 给我们，为我们提供了极其丰富的、极具价值的各类信息和建议。2003 年 1 月至 4 月，我们对参加全国计算机等级考试的部分考生进行了历时四个月的联合调查*，本次调查共计 3756 人，按地理区域划分涉及华北（29%）、华中（23%）、华南（17%）、东北（14%）、西南（11%）、西北（6%）。

从本次联合调查的结果分析中，我们得到了很多极具价值的结论，其中以下几类问题比较集中：

- 76% 的被访者认为笔试部分相对上机部分较为容易，13% 认为上机部分相对容易，11% 认为无法比较。可见大部分被访者认为“笔试部分通过考试相对容易，而上机部分较难考过”。
- 63% 的被访者对目前市场上的辅导资料不满意，主要表现在错误多、内容严重重复、价格贵等方面。其中对市场上充斥着许多盗印的教材也有反映。
- 82% 的被访者希望能在复习阶段进行模拟上机训练，以适应上机考试的特殊环境。

为了更好地配合广大考生的考前复习，同时根据联合调查的部分结论，我们新推出《全国计算机等级考试——上机习题集》，特点如下：

☆ 上机考试环境演示，真实考试环境演示，每一步均有提示，以便考生提前适应考试环境。

☆ 提供了大量与真实考试难度及题型相当的考题供考生训练。每套题都配有参考答案，可使考生针对性的进行练习，提高上机考试通过率。

☆ 和教材吻合，突出重点难点。

本光盘（三级 C 语言程序设计）的序列号为：ADH3R-SDFAS-69959-HADTA 进行模拟练习时，请勿脱离光盘使用。

说明：三级 C 语言程序设计适用于三级网络技术、三级数据库技术、三级信息管理技术，本光盘以三级网络技术的界面为例。

* 本次计算机等级考试联合调查的样本从数量上分析具有一定的普遍代表性，但由于地域、受访者职业分布、问卷设计缺陷及调查途径等因素的限制，调查也存在一定的样本缺陷性，以上结论请慎重使用。

第一部分 上机题解

1. 考点综述

上机考试是计算机等级考试必不可少的一部分,考试时间 60 分钟。评分标准是依据程序得出的答案和标准答案相比较,看答对的百分比给分。成绩分为优、良、中、及格和不及格。它的重要性不言而喻,所以大家必须注意。对于上机考试,大家要注意以下一些问题。

(1) 平时要多做上机练习。有些人平时学习 C 语言自我感觉还可以,可是一上机编程就发蒙了,所以说上机练习是复习三级考试必不可少的一个环节。

(2) 要熟悉考试环境。就是考试用什么样的软件,什么界面,怎样操作,我们在光盘中作了详细介绍。平时可以用 UC DOS 和 TC2.0 来模拟,或者现在有很多模拟上机考试的软件,都可以来做一做。有的同志平时用 VC 调试程序,这样有时候会在考试的时候不知道怎样操作,影响考试的进行。

(3) 对于许多的 C 语言的语句,如果嫌它太多太烦也没关系,可以试一试 TC2.0 的帮助系统。这个对于英语学的比较好的很适用。

(4) 在考试时,不要急于编程,一定要先想好如何做,再一步一步去实现想法。有许多同志一看到题目就马上写程序,结果写了一大片才发现这条路根本走不通,于是删了又改,心浮气躁怎么可以编出很好的程序。

(5) 学习一些 DOS 命令,由于上机考试环境在 DOS 下,所以事先熟悉 DOS 的一些操作可以使你考试时更加如鱼得水,轻松取胜。

(6) 在编程时,一定要注意“=”和“==”的区别,前者是赋值运算符,后者是用来比较的,才是我们普通意义上的等于。由于思维的定势,很多人都会在判断依据中使用“=”,造成不该有的错误。所以再写完 if 或者 for 循环等语句时一定要回头检查一下,是不是“==”写错了。

(7) 对于 if……else 语句和 for 循环语句一定要熟练,同时对于字符和字符串的概念和操作也很重要。几乎每次考试都会涉及以上的内容。

(8) 注意题目已经给出的数据是否已经初始化,不要轻易的认为数据在定义后的初值就是 0,其实对于大多数操作系统中,这个值是一个随机数。所以不能对未初始化的数据直接引用。如:

```
int sum;
```

```
sum++;
```

程序中 sum 的数值是不可预知的,从而引起程序差错。对于数组要对每个元素赋初值 0。

(9) 上机打完题目后,要记着看看你程序的输出文件。有的时候,由于本身考题的问题或者系统的问题,有的考生程序作对了,可是输出文件却是错的,最后发现是源程序中的写入文件的地方错了。所以大家要注意看一看,以免发生不必要的麻烦。

2. 考题精解

【例 1】下列程序的功能是:已知学生的记录由学号和学习成绩构成, N 名学生的数据已存入 a 数组中。找出成绩最低的学生记录(假定最低成绩的记录中唯一的), 通过形参返回。请考生编写函数 mmm(STU a[],

STU *s)实现程序的要求,最后调用函数 readwriteDAT()把结果输出到文件 e1.out 中。

例如: KS01 87

KS09 97

KS11 67

则调用该函数后,输出 The lowest : KS09, 67

注意:部分源程序已经给出。

请勿改动主函数 main()和输出数据函数 readwriteDAT()的内容。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define N 10
void readwriteDAT();
typedef struct ss
{ char num[10]; int s; } STU;
void mmm(STU a[], STU *s)
{
    main()
    {
        STU a[N]={ {"A01",81}, {"A02",89}, {"A03",66}, {"A04",87}, {"A05",77},
                    {"A06",90}, {"A07",79}, {"A08",61}, {"A09",80}, {"A10",71} };
        STU m;
        int i;
        clrscr();
        printf("***** The original data *****\n");
        for(i=0; i<N; i++)
            printf("No = %s Mark = %d\n", a[i].num, a[i].s);
        mmm(a, &m);
        printf("***** THE RESULT *****\n");
        printf("The lowest : %s , %d\n", m.num, m.s);
        readwriteDAT();
    }
void readwriteDAT()
{
    FILE *rf, *wf;
    STU a[N], m;
    int i;

    rf = fopen("e1.in", "r");
    wf = fopen("e1.out", "w");
    for(i = 0 ; i < 10; i++)
        fscanf(rf, "%s %d", a[i].num, &a[i].s);
```

```

mmm(a, &m) ;
fprintf(wf, "The lowest : %s, %d\n", m.num, m.s);
fclose(rf) ;
fclose(wf) ;
}

```

答案:

```

void mmm(STU a[], STU *s)
{
    int i;
    STU b;
    b = a[0];

    for (i=1; i<N; i++)
        if (a[i].s < b.s)
            b = a[i];

    (*s).s = b.s;
    for(i=0; i<10; i++)
        (*s).num[i] = b.num[i];
}

```

解析: 本题主要考查对结构体引用的掌握和排序的基本知识。

结构体数据结构是将不同类型的数据组合成一个有机的整体, 便于引用, 这些组合在一个整体中的数据是互相联系的。在对结构体类型变量引用时要注意:

(1) 不能将一个结构体变量作为一个整体进行输入和输出。如: “fprintf(wf, “The lowest : %s, %d\n”, m);” 是错误的, 必须对每个成员逐步操作。

(2) 对成员变量可以像普通变量一样进行各种运算。如: (*s).num[i]++

(3) 对结构体成员变量的引用可以是“.”成员(分量)运算符, 也可以用指向运算符“->”, (*p).num 和 p->num 等价。注意这里的 p 是指向结构体变量的指针。

(4) 成员(分量)运算符“.”在所有的运算符中优先级最高。要注意优先级的问題, 如: (*s).num[i] 不要写成 *s.num[i], 这样就会出错。

【例 2】函数 ReadDat() 实现从文件 E2.IN 中读取一篇英文文章, 存入到字符串数组 xx 中; 请编制函数 StrJL() 分别按行删除所有的空格、数字、标点符号等非字母字符, 并对每行的单词按升序排序后仍存放回数组 xx 中。最后调用函数 WriteDat() 把结果输出到文件 E2.OUT 中。

原始数据文件存放格式: 文章小于 50 行; 每行宽度小于 80 字符, 含标点符号和空格; 每个单词小于 20 个字符。

注意: 部分源程序已经给出。

请勿改动主函数 main()、读数据函数 ReadDat() 和输出数据函数 WriteDat() 的内容。

```

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

```

```

char xx[50][80] ;
int maxline = 0 ; /* 文章的总行数 */

```

```
int ReadDat(void) ;
void WriteDat(void) ;
```

```
void StrJL(void)
{
```

```
void main()
{
```

```
clrscr() ;
if(ReadDat()) {
    printf("数据文件 E2.IN 不能打开!\n\007") ;
    return ;
}
```

```
StrJL() ;
WriteDat() ;
}
```

```
int ReadDat(void)
```

```
{
    FILE *fp ;
```

```
    int i = 0 ;
```

```
    char *p ;
```

```
    if((fp = fopen("e2.in", "r")) == NULL) return 1 ;
    while(fgets(xx[i], 80, fp) != NULL) {
```

```
        p = strchr(xx[i], '\n') ;
```

```
        if(p) xx[i][p - xx[i]] = 0 ;
```

```
        i++ ;
```

```
    }
```

```
    maxline = i ;
```

```
    fclose(fp) ;
```

```
    return 0 ;
}
```

```
void WriteDat(void)
```

```
{
```

```
    FILE *fp ;
```

```
    int i ;
```

```
    fp = fopen("e2.out", "w") ;
```

```
    for(i = 0 ; i < maxline ; i++) {
```

```
        printf(" %s\n", xx[i]) ;
```

```
        fprintf(fp, " %s\n", xx[i]) ;
```

```
    }
```

```
    fclose(fp) ;
```

```
}
```

答案:

```

void StrJL(void)
{
    int i,j=0,k,l,m;
    char str0[21],ww[80][20],*p;

    for(i=0;i<maxline;i++)
    {
        m=0;j=0;
        for(k=0;k<=strlen(xx[i]);k++)
            /*判断是否是字母*/
            if(isalpha(xx[i][k]))
            {
                str0[m]=xx[i][k];
                m++;
            }
        else
        {
            str0[m]='\0';
            if (str0[0]!='\0')
            {
                strcpy(ww[j],str0);j++;
                m=0;str0[m]='\0';
            }
        }
        /* 排序 */
        for(k=0;k<j;k++)
        {
            for(l=0;l<j;l++)
            {
                if(strcmp(ww[k],ww[l])>0)
                {
                    strcpy(str0,ww[k]);
                    strcpy(ww[k],ww[l]);
                    strcpy(ww[l],str0);
                }
            }
        }
        strcpy(xx[i], "");
        for(k=0;k<j;k++)
            strcat(xx[i],ww[k]);
    }
    /*第一个 for 循环终止*/
}

```

① 字符串处理函数

② 循环

答案

解析: 本题首先按行从句子中分离出单词,再将各单词存入一个临时的字符串数组,最后对此字符串变量排序,结果写回 xx 数组。函数中 str0 是用来存放每个单词。当读入字符不是字母时,即 isalpha 返回 0 时,说明本单词已经结束,写入到 ww 数组中。排序采用冒泡排序法。注意 strcmp(str1, str2) 函数的返回值:当 str1<str2 时返回负数;str1>str2 时返回正数;相等时返回 0。最后利用 strcat 函数将结果写入 xx 数组。

本题难度偏大,要注意字符串函数的使用。

【例 3】 写两个函数, hcf() 求两个整数的最大公约数, lcd() 求两个整数的最小公倍数。

注意: 部分源程序已经给出。

```

#include "stdio.h"
#include "math.h"

hcf(int u,int v)
{
}

lcd(int u,int v,int h)
{
}

main()
{
    int u,v,h,l;
    printf("input 2 integers:");
    scanf("%d %d",&u,&v);
    h=hcf(u,v);
    printf("H.C.F=%d\n",h);
    l=lcd(u,v,h);
    printf("L.C.D=%d\n",l);
}

```

答案:

```

hcf(int u,int v)
{
    int a,b,t,r;
    if(u>v)
    {
        t=u;u=v;v=t;
    }
    a=u;b=v;
    while((r=b%a)!=0)
    {
        b=a;a=r;
    }
    return a;
}

lcd(int u,int v,int h)
{
    return (u*v/h);
}

```

解析: 用“辗转相除法”，求最大公约数用较大的数模除较小的数（即求余数）。若模为 0，则较小的数即为最大公约数；否则，将较小的数置为较大数，将求得的模作为较小的数，继续上述过程。直到模为 0，则最后的较小数即为最大公约数。求最小公倍数时用两个数的乘积除以它们的最大公约数即可。

【例 4】 下列程序的功能是：统计在 `tt` 字符串中数字 0~9 各自出现的次数，并存入 `pp` 数组。请考生编写函数 `cnt(char *tt, int pp[])` 实现程序的要求，最后调用函数 `readwriteDAT()` 把结果输出到文件 `e4.out` 中。

例如：当输入字符串：`abc1124def5gabcdeabc` 后，输出的

结果应该是:

0210110000

注意: 部分源程序已经给出。

请勿改动主函数 main() 和输出数据函数 readwriteDAT() 的内容。

```
#include <conio.h>
#include <stdio.h>
void readwriteDAT();
int pp[26];
```

```
void cnt(char *tt, int pp[])
{
```

```
main()
```

```
{
    char tt[1000];
    int k, n;
    clrscr();
    printf("\nPlease enter a char string:");
    scanf("%s", tt);
    cnt(tt, pp);
    for (k = 0; k < 10; k++)
        printf ("%d ", pp[k]);
    printf("\n");
    readwriteDAT();
}
```

```
void readwriteDAT()
```

```
{
    int k, n, i;
    FILE *wf;

    wf = fopen("e4.out", "w");
    if(!wf)
        return;
    for(k = 0; k < 10; k++)
    {
        fprintf(wf, "%d ", pp[k]);
        printf("%d ", pp[k]);
    }
    fprintf(wf, "\n");
    printf("\n");
    fclose(wf);
}
```

答案:

```
void cnt(char *tt, int pp[])
{
```

```

char *digit="0123456789";
int i,j;
for(i=0;i<strlen(tt);i++)
{
    j=0;
    for(j=0;j<strlen(digit);j++)
        if (tt[i]==digit[j])
        {
            pp[j]++;
            break;
        }
}

```

*Strlen(tt) * 10
= 10000 * 10 = 100000*

解析：本题主要考查对字符串的操作。在判断字符串长度时，我们可以用库函数 `strlen`，还可以利用字符串最后字节为 '\0' 的性质来判断，如：`for (i=0; *(tt+i)!='\0'; i++)`。

同时我们可以利用字符类型和整型之间的对应关系。如：我们想判断 `str` 是否是大写字母，可以用 `ascii` 码：`((str>=65)&&(str<=90))`；也可以直接写为：`((str>='A')&&(str<='Z'))`。这样在简单的同时，也增强了程序的可读性。

【例 5】已知数据文件 `in.dat` 中存有 200 个四位数，并已调用读函数 `ReadDat()` 把这些数据存入数组 `a` 中，请编制函数 `jsValue()`，其功能是：如果千位数上的数加个位数上的数等于十位数上的数加百位数上的数，则统计出满足此条件四位数的个数 `cnt`，并把这些数按从大到小的顺序存入数组 `b` 中，最后调用 `WriteDat()` 把结果输出到 `out.dat` 文件中。

注意：部分源程序已经给出。

请勿改动主函数 `main()`、读数据函数 `ReadDat()` 和输出数据函数 `WriteDat()` 的内容。

```

#include "stdio.h"
#define MAX 200
int a[MAX],b[MAX],cnt=0;

jsValue()
{

}

main()
{
    int i;
    if(ReadDat())
        return;
    jsValue();
    writeDat();
}

ReadDat()
{
    FILE *fp;
    int i;
    fp=fopen("in.dat","r");

```

```

if(!fp)
{
    printf("open file error!\n");
    return 1;
}
for(i=0;i<MAX;i++)
    fscanf(fp,"%d",&a[i]);
fclose(fp);
}
writeDat()
{
    FILE *fp;
    int i;
    fp=fopen("out.dat","w");
    fprintf(fp,"%d\n",cnt);
    printf("%d\n",cnt);
    for(i=0;i<cnt;i++)
    {
        fprintf(fp,"%d\n",b[i]);
        printf("%d\n",b[i]);
    }
    fclose(fp);
}

```

答案:

```

jsValue()
{
    int i,gw,sw,bw,qw,j,k;
    for(i=0;i<MAX;i++)
    {
        gw=a[i]%10;
        sw=(a[i]%100)/10;
        bw=(a[i]/100)%10;
        qw=a[i]/1000;
        if(qw+bw==gw+sw)
        {
            for(j=0;j<cnt;j++)
                if(a[i]<b[j])
                {
                    for(k=cnt-1;k>=j;k--)
                        b[k+1]=b[k];
                    b[j]=a[i];
                    cnt++;
                    break;
                }
            if(j==cnt)
            {
                b[cnt]=a[i];
                cnt++;
            }
        }
    }
}

```

Handwritten notes:

- $j < cnt; j++$
- $k = cnt - 1; k = j$
- $k = 0$
- 1 2 5
- 2

解析：本题主要考查对数字的操作和排序。

题中 gw、sw、bw、qw 分别表示个位、十位、百位、千位。它们的算法一般都是求模和整除，如题中所示。还可以有其它的方法来求：比如：

$bw = a[i] \% 10 / 100;$ 或： $bw = (a[i] - qw * 1000) / 100;$

排序也可以先存储然后再用固定的算法（如：冒泡法）来求得结果。希望大家可以多多练习，以便在答题的时候可以很灵活的运用已有的知识。

```

if(!fp)
    printf("open file error!\n");
return 1;
fclose(fp);

FILE *fp;
int i;
fp=fopen("out.dat","w");
fprintf(fp,"%d\n",cnt);
printf("%d\n",cnt);
for(i=0;i<cnt;i++)
{
    printf(fp,"%d\n",b[i]);
    printf("%d\n",b[i]);
}
fclose(fp);
}

// 答案:
jswalues()
{
    int i,qw,sw,bw,dw;
    for(i=0;i<MAX;i++)
    {
        gw=a[i]%10;
        sw=a[i]/10%10;
        bw=(a[i]/100)%10;
        qw=a[i]/1000;
        if(qw+sw==dw+bw)
        {
            for(j=0;j<cnt;j++)
            {
                if(a[j]<0)
                {
                    for(k=cnt-1;k>=j;k--)
                    {
                        p[k+1]=p[k];
                        a[j]=a[k];
                        cnt++;
                        break;
                    }
                }
            }
        }
    }
}

```

$$A_n = A_{n-2} + 2A_{n-1}$$

{

2. 函数 ReadDat()实现从文件 ENG.IN 中读取一篇英文文章,存入到字符串数组 xx 中;请编制函数 encryptChar(), 按行优先把所有的小写字母替换成它前面的字母其它不变, a 换成 z, 例如: Aabm.() 输出

Azal() 最后调用函数 WriteDat()把结果 xx 输出到文件 out.DAT 中。部分源程序已给出, 原始数据文件存放的格式是: 每行的宽度均小于 80 个字符。请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<ctype.h>
```

```
unsigned char xx[50][80];
```

```
int maxline=0; /*文章的总行数*/
```

```
int ReadDat(void);
```

```
void WriteDat(void);
```

```
void encryptChar()
```

```
{
```

```
}
```

```
void main()
```

```
{
```

```
clrscr();
```

```
if(ReadDat())
```

```
{
```

```
printf("数据文件 ENG.IN 不能打开! \n\007");
```

```
return;
```

```
}
```

```
encryptChar();
```

```
WriteDat();
```

```
}
```

```
int ReadDat(void)
```

```
{
```

```
FILE *fp;
```

```
int i=0;
```

```
unsigned char *p;
```

```
if((fp=fopen("eng.in","r"))==NULL)
```

```
return 1;
```

```
while(fgets(xx[i],80,fp)!=NULL)
```

```
{
```

```

        p=strchr(xx[i],'\n');
        if(p)
            *p=0;
        i++;
    }
    maxline=i;
    fclose(fp);
    return 0;
}

```

```

void WriteDat(void)
{
    FILE *fp;
    int i;

    fp=fopen("out.dat","w");
    for(i=0;i<maxline;i++)
    {
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

☆☆

3. 函数 ReadDat()的功能是实现从文件 IN.DAT 中读取一篇英文文章存入到字符串数组 xx 中; 请编制函数 StrCharJR();该函数的功能是: 以行为单位把字符串中的所有字符的 ASCII 值右移 4 位, 然后把右移后的字符 ASCII 值再加上原字符的 ASCII 值, 得到新的字符仍存入原字符串对应的位置上。最后把已处理的字符串仍按行重新存入字符串数组 xx 中, 最后调用函数 WriteDat()把结果 xx 输出到文件 OUT.DAT 中。

原始数据文件存放的格式是: 每行的宽度均小于 80 个字符, 含标点符号和空格。

注意: 部分源程序已给出。

请勿改动主函数 main()、读数据函数 ReadDat()和输出数据函数 WriteDat()的内容。

```

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

```

```

char xx[50][80];
int maxline=0;/*文章的总行数*/

```

```

int ReadDat(void);
void WriteDat(void);

```