

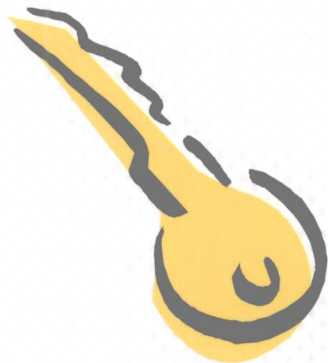


全国 计算机等级考试 过关 宝典

——三级C语言上机习题解析

翟自强 马志强 主编

- ◆ 按照最新大纲 紧扣考试重点
- ◆ 精心编排内容 知识易懂易记
- ◆ 大量模拟习题 细致全面解析
- ◆ 真题实战热身 助你顺利过关



天津大学出版社
TIANJIN UNIVERSITY PRESS

内容简介

本书是“全国计算机等级考试过关宝典”系列中的一本，旨在帮助考生快速掌握三级C语言上机考试的解题技巧。本书内容紧扣考试大纲，精选历年真题，详细解析了各种题型，包括程序填空题、程序改错题、程序运行题、程序填空题、程序改错题、程序运行题等。本书可作为考生备考的参考书，也可作为计算机专业及相关专业的教材。

全国计算机等级考试过关宝典—— 三级 C 语言上机习题解析

翟自强 马志强 主编

王宝和 巩国忠 副主编

吴迪 张振河
李玉石 李红英

中国版本图书馆CIP数据核字(2009)第062077号

出版人：马志强
地址：天津市河西区
电话：(022) 2333 3333
网址：www.tjupress.com
邮编：300070
全套：1册
本册：182mm×260mm
开本：1/16
印张：288
字数：200千字
版次：2009年4月第1版
印次：2009年4月第1次
印数：1-4000
定价：25.00元



天津大学出版社

TIANJIN UNIVERSITY PRESS

本书经教育部考试中心审定，可作为全国计算机等级考试教材。

天津出版

天津出版

内容简介

本书紧密结合最新考试大纲,采用上机题库习题训练与应试指导相结合的方法进行编排。书中对三级 C 语言上机考试涉及的重点、难点进行分析,将常见类型题目和较难的题目做了详细讲解,指导考生全面掌握考试要点。本书将三级 C 语言上机考试题库中的习题分成十四类进行编写,且每一题型的代码统一,便于考生记忆。通过练习,考生既可达到学习的目的,也可自测学习效果。

本套书共五册,可同时作为大专院校及计算机等级考试培训班学生用书。

图书在版编目(CIP)数据

全国计算机等级考试过关宝典. 三级 C 语言上机习题
解析/翟自强,马志强主编. —天津:天津大学出版社,
2009.4

ISBN 978-7-5618-3022-2

I. 全… II. ①翟…②马… III. ①电子计算机—水平
考试—自学参考资料②C 语言—程序设计—水平考试—自
学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字(2009)第 065077 号

出版发行 天津大学出版社
出 版 人 杨欢
地 址 天津市卫津路 92 号天津大学内(邮编:300072)
电 话 发行部:022-27403647 邮购部:022-27402742
网 址 www.tjup.com
印 刷 廊坊市长虹印刷有限公司
经 销 全国各地新华书店
开 本 185mm×260mm
印 张 11.5
字 数 288 千
版 次 2009 年 4 月第 1 版
印 次 2009 年 4 月第 1 次
印 数 1-4 000
定 价 22.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

前 言

随着计算机科学的高速发展和计算机技术的广泛应用,各种专业人才掌握计算机知识和学会使用计算机已成当务之急。既掌握一定的专业技术,又具备计算机应用能力的人员越来越受到用人单位的重视和欢迎。

国家教育部考试中心推出的“全国计算机等级考试”是面向各个阶层、各个行业,不论年龄、专业和知识背景,统一、公正、科学的社会化考试,目的是以考促学,向社会推广普及计算机知识和应用,为选拔人才提供科学依据。现在,参加这项考试的考生人数已累计达 500 万人,其中有 180 多万人获得了各级证书。实践证明,这个同工作岗位培训密切相关的社会证书考试,促进了不同群体对计算机应用技术的掌握和普及,并且为用人单位提供了权威性的水平认证,也同样受到了广大考生的欢迎和用人单位的广泛认可。为了进一步保证这项考试公平、公正,真实地考核出考生的计算机操作水平和编程能力,教育部考试中心决定从 2001 年秋季开始,实行新的上机题库,已由全国计算机等级考试委员会编写出版了《全国计算机等级考试上机考试习题集》,供考生考前学习使用。

本书就是与该习题集中《三级 C 语言上机考试习题集》配套的参考书。书中对题库中的每一道题,力求用通俗易懂的编程方法给出正确的答案,对不同类型题做了详细解析,方便读者脱机记忆。这些题已经上机调试通过。该书应试导向准确、针对性强。考生通过实战练习,就能在较短时间内巩固所学知识,掌握要点,突破难点,把握考点,熟练掌握答题方法和技巧,适应考场氛围,并最终顺利通过考试。

本书参编人有于筱荔、张丹阳、李强、冯光、柴君、朱云霞。

特别感谢孙桂玲、耿福建、朱峰、周一林、郑爱华、潘阳、符继全、谢福宁、李雪宾、王萍、路文悦、祁兰、宣红晶、李栋、周国瑞、刘晶、王浩洳参与本书各题上机验证核对工作。

由于时间仓促,不足之处在所难免,特别是每道题的答案不一定是最佳答案,恳请广大读者批评指正。

编者

2009 年 3 月

目 录

第一部分 三级 C 语言上机试题题型分析	1
一、替换字符	1
二、字符串左右排序和比较	3
三、正整数排序求平均值(包括将数拆散、求最大最小值)	5
四、产品结构体问题的操作	6
五、素数	9
六、数字排序	10
七、其他数学计算	12
八、数字或字符移位后的计算	13
九、学生成绩结构体问题的操作	15
十、字符串(单词)的倒置和删除	16
十一、选票问题	19
十二、出圈问题	21
第二部分 三级上机题库分类 100 题	23
一、结构体数据排序	23
二、数位对拆和结果排序	28
三、数位对拆和统计均值	60
四、特值统计	66
五、统计方差	69
六、销售记录排序	74
七、选票问题	91
八、正数统计和特值处理	96
九、字符串处理	102
十、字符串的左右排序	105
十一、字符取余转换	110
十二、字符移位处理	126
十三、字符转换和排序	132
十四、杂类	141
附录 1 计算机等级考试 VC++6.0 上机指导	160
附录 2 考试系统简介	168
附录 3 三级 C 语言上机考试操作步骤	173

第一部分 三级 C 语言上机 试题题型分析

三级网络技术的 C 语言上机试题难度较大,题型多样,因此颇难掌握。现将这些试题分类解析。

三级的上机题可以分为十几种类型,每类中的题目数量不一,多则十几道题,少则三四道题,但是每类题都差不多,每类中只要掌握一道题,再理清做题的思路,就相当于全掌握了。

一、替换字符

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。请编写函数 encryptChar(),按给定的替代关系对数组 xx 中的所有字符进行替代,仍存入数组 xx 的对应的位置上,最后调用函数 WriteDat()把结果 xx 输出到文件 PS10.DAT 中。

替代关系: $f(p) = p * 11 \bmod 256$ (p 是数组中某一个字符的 ASCII 值, $f(p)$ 是计算后新字符的 ASCII 值)。如果原字符的 ASCII 值是偶数或计算后 $f(p)$ 值小于等于 32,则该字符不变,否则将 $f(p)$ 所对应的字符进行替代。

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;

```

```

        maxline=i;

```

```

        fclose(fp);

```

```

        return 0;

```

```

void WriteDat(void)
{

```

```

    FILE *fp;

```

```

    int i;

```

```

    fp=fopen("ps10.dat","w");

```

```

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

```

```

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

```

```

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

```

```

    }
    fclose(fp);
}

```

注:在 ReadDat() 函数中,由于 fgets() 函数读入数据时没有读入字符串结束符 '\0', 因而用 while() 循环在 xx 数组每一行末尾将换行符 '\n' 替换成结束符 '\0'。

参考答案:

```

void encryptChar()
{

```

```

    int i,j,t;

```

```

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

```

```

    }
    for(j=0;j<strlen(xx[i]);j++)
    {
        t=xx[i][j]*11%256;
        if(t<=32 || xx[i][j]%2==0) continue;
        xx[i][j]=t;
    }
}

```

解题思路:

这类题比较简单,每道题的不同之处在于要替换字符的条件不一样,但条件中总会说“如果该字符符合什么条件,字符就不变,否则就替换”。解决这个问题最省事的办法就是把不变的条件写好,前面加上逻辑非,如 $!(xx[i][j]\%2==0 \parallel fp \leq 32)$ 。

另一个要注意的问题是把改变后的字符临时存放在一个变量中。这个变量(如本题中的 fp)必须是 unsigned char 类型或 int 类型的变量。若写成 char 类型的会出错,因为从公式 $fp = xx[i][j] * 11 \% 256$ 中可以看出 fp 的取值范围应在 0 ~ 255 之间,而 char 类型的变量范围仅在 -128 ~ +127 之间。

在本题中用两重循环遍历字符串数组,外层循环遍历行, maxline 中存放的是 xx 数组中存放字符串的真实的行数;内层循环遍历每行的每个字符,因为在 xx 中每行是一个字符串,而每个字符串都有一个 '\0' 做结束标志,所以内层循环的表达式 2 用 $xx[i][j] != '\0'$ 控制循环。

二、字符串左右排序和比较

函数 ReadDat() 实现从文件 in.dat 中读取 20 行数据存放到字符串数组 xx 中(每行字符串长度均小于 80)。请编制函数 jsSort(), 功能是以行为单位对字符串按给定的条件进行排序,排序后的结果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。

条件:从字符串中间一分为二,左边部分按字符的 ASCII 值升序排序,排序后左边部分与右边部分进行交换。如果原字符串长度为奇数,则最中间的字符不参加处理,字符仍放在原位置上。

位置	1	2	3	4	5	6	7	8	9
例如:原字符串	d	c	b	a	h	g	f	e	
	4	3	2	1	9	8	7	6	5
则处理后字符串	h	g	f	e	a	b	c	d	
	8	7	6	5	9	1	2	3	4

部分源程序已给出。

请勿改动主函数 main()、读函数 ReadDat() 和写函数 WriteDat() 的内容。

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
char xx[20][80];

```

```
void jsSort()
```

```
{
```

```
}
```

```
void main()
```

```
{
```

```
    readDat();
```

```
    jsSort();
```

```
    writeDat();
```

```
}
```

思路分析

本题要求对读入的字符串进行排序，排序后的字符串再写入文件。排序算法有多种，这里采用冒泡排序法。冒泡排序法的基本思想是：将待排序的数组中相邻的两个数进行比较，如果前者大于后者，则交换它们的位置。重复这一过程，直到整个数组有序为止。

```
FILE * in;
```

int i=0; 这里 i 表示当前要比较的字符串的索引，从 0 开始。

char * p; 这里 p 表示当前字符串的指针，指向字符串的起始位置。

in = fopen("in.dat", "r"); 打开输入文件 in.dat，以只读方式打开。

```
while(i < 20 && fgetc(xx[i], 80, in) != NULL)
```

这里 i 表示当前要比较的字符串的索引，从 0 开始。

p = strchr(xx[i], '\n'); 这里 p 表示当前字符串的换行符的位置。

if(p) * p = 0; 这里将换行符替换为字符串结束符 '\0'。

```
i++;
```

这里 i 表示当前要比较的字符串的索引，从 0 开始。

writeDat(); 将排序后的字符串写入文件 out.dat。

fclose(in); 关闭输入文件 in.dat。

writeDat(); 将排序后的字符串写入文件 out.dat。

```
writeDat()
```

这里 i 表示当前要比较的字符串的索引，从 0 开始。

FILE * out; 这里 out 表示输出文件的指针。

int i; 这里 i 表示当前要比较的字符串的索引，从 0 开始。

```
out = fopen("out.dat", "w");
```

这里 out 表示输出文件的指针。

```
for(i=0; i < 20; i++) {
```

这里 i 表示当前要比较的字符串的索引，从 0 开始。

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

这里 printf 用于输出字符串 xx[i]。

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

这里 fprintf 用于将字符串 xx[i] 写入文件 out。

```
}
```

这里 fprintf 用于将字符串 xx[i] 写入文件 out。

```
fclose(out);
```

这里 fclose 用于关闭输出文件 out。

```
}
```

这里 fclose 用于关闭输出文件 out。

注意：先采用冒泡法对左边部分进行升序排序，然后将排序后的左半部与右半部按对应位进行调换。

参考答案：

```
void jsSort()
```

```
#include <string.h>
```

```
#include <conio.h>
```

```
char xx[20][80];
```

```

int i, strl, half, j, k;
char ch;
for(i = 0; i < 20; i++) /* 行循环 */
{ strl = strlen(xx[i]); /* 每行长度 */
  half = strl/2;
  for(j = 0; j < half - 1; j++) /* 每行的第 j 个位置 */
  for(k = j + 1; k < half; k++)
  if(xx[i][j] > xx[i][k])
  { ch = xx[i][j]; /* 每次将最小数赋给 xx[i][j] */
    xx[i][j] = xx[i][k];
    xx[i][k] = ch;
  }
  for(j = half - 1, k = strl - 1; j >= 0; j--, k--)
  { ch = xx[i][j]; xx[i][j] = xx[i][k]; xx[i][k] = ch; }
}
}

```

三、正整数排序求平均值(包括将数拆散、求最大最小值)

已知数据文件 in. Dat 中存有 300 个四位数,并已调用读函数 readDat()把这些数存入数组 a 中,请编制一函数 jsValue(),功能是求出千位数上的数加个位数等于百位数上的数加十位数上的数的个数 cnt,再求出所有满足此条件的四位数的平均值 pjz1 以及不满足此条件的四位数的平均值 pjz2,最后调用写函数把结果输出到 out. Dat 文件。

例如:6712, $6 + 2 = 7 + 1$,则该数满足条件,计算平均值 pjz1,且个数 $cnt = cnt + 1$ 。8129, $8 + 9 < 1 + 2$,则该数不满足条件,计算平均值 pjz2。

部分源程序已给出。

程序中已定义数组: a[300],已定义变量: cnt, pjz1, pjz2。

请勿改动主函数 main()、读函数 readDat()和写函数 writeDat()的内容。

```

#include <stdio.h>
int a[300], cnt = 0;
double pjz1 = 0.0, pjz2 = 0.0;
jsValue()
{
    for(i = 0; i < 300; i++)
    {
        int g = a[i] % 10;
        int s = a[i] / 10 % 10;
        int b = a[i] / 100 % 10;
        int q = a[i] / 1000 % 10;
        if((q + g) == (s + b)) cnt++;
        else cnt--;
    }
}
main()
{
    readDat();
    jsValue();
    writeDat();
}

```

```

printf("cnt = %d\n 满足条件的平均值 pjz1 = %7.21f\n 不满足条件的平均值 pjz2\n\n", cnt, pjz1, pjz2);
}

readDat()
{
    FILE *fp;
    int i;
    fp = fopen("in.dat", "r");
    for(i=0; i<300; i++) fscanf(fp, "%d", &a[i]);
    fclose(fp);

    writeDat()
    {
        FILE *fp;
        fp = fopen("out.dat", "w");
        fprintf(fp, "%d\n%7.21f\n%7.21f\n", cnt, pjz1, pjz2);
        fclose(fp);
    }

    jsValue()
    {
        int i, g, s, b, q, k=0;
        for(i=0; i<300; i++)
        {
            g = a[i] % 10;
            s = a[i] % 100 / 10;
            b = a[i] / 100 % 10;
            q = a[i] / 1000;
            if((q + g) == (s + b)) { cnt++; pjz1 += a[i]; }
            else { k++; pjz2 += a[i]; }
        }
        pjz1 /= cnt;
        pjz2 /= k;
    }
}

```

四、产品结构体问题的操作

本类型题共出现十道题,题目说明很相似,只是排序的条件不同,所以这种题的代码非常

规则,全部采用冒泡法排序,只改以下排序的条件即可,故把这类题称为“按金额、代码排序题”。

例如:已知在文件 in. Dat 中存有 100 个产品销售记录,每个产品销售记录由产品代码 dm (字符型 4 位)、产品名称 mc(字符型 10 位)、单价 dj(整型)、数量 sl(整型)、金额 je(长整型)四部分组成。其中:金额 = 单价 * 数量。函数 readDat()是读取这 100 个销售记录并存入结构数组 sell 中。请编写函数 sortDat(),功能是按产品名称从大到小进行排列,若产品名称相同,则按金额从大到小进行排列,最终排列结果仍存入结构数组 sell 中,最后调用函数 WriteDat()把结果输出到文件 OUT8. DAT 中。

解题思路:

这类题一律用冒泡法排序,但注意冒泡法排序要使用交换变量。这里的交换变量类型应和被排序的数组的类型一致,本类题中应使用结构体类型的变量。本题中的产品名称和产品代码都为字符串。对其进行比较时,应使用 strcmp 函数。本类题排序都用两个关键字,分为第一关键字(如上题中的“产品名称”)和第二关键字(如上题中的“金额”)。这种排序都是先按第一关键字排序,如果第一关键字相同,再按第二关键字排序,这时即可以写成两个排序过程,也可以写在一起。为了简略,把两个排序写成一个排序过程,不过这时排序的条件稍有些复杂,应该注意。排序条件由两部分构成,请注意看程序中的括号。题目说明中黑体部分就是本类题经常变化的地方。

排序程序如下:

```
#include <stdio. h >
#include <string. h >
#include <conio. h >
#include <stdlib. h >
#define MAX 100
typedef struct
{
    char dm[5]; /* 产品代码 */
    char mc[11]; /* 产品名称 */
    int dj; /* 单价 */
    int sl; /* 数量 */
    long je; /* 金额 */
} PRO;
PRO sell[MAX];
void ReadDat();
void WriteDat();
void SortDat()
{
}

void main()
{
```

```

memset( sell,0,sizeof( sell));
ReadDat();
SortDat();
WriteDat();
void ReadDat()
{
    FILE *fp;
    char str[80],ch[11];
    int i;
    fp = fopen("IN. DAT","r");
    for(i=0;i<100;i++){
        fgets(str,80,fp);
        memcpy( sell[i]. dm, str,4);
        memcpy( sell[i]. mc, str+4,10);
        memcpy( ch, str+14,4);ch[4]=0;
        sell[i]. dj = atoi( ch);
        memcpy( ch, str+18,5);ch[5]=0;
        sell[i]. sl = atoi( ch);
        sell[i]. je = (long)sell[i]. dj * sell[i]. sl;
    }
    fclose( fp);
}

void WriteDat( void)
{
    FILE *fp;
    int i;
    fp = fopen("OUT8. DAT","w");
    for(i=0;i<100;i++){
        fprintf( fp,"%s %s %4d %5d %10Ld\n", sell[i]. dm,sell[i]. mc,sell[i]. dj,
            sell[i]. sl,sell[i]. je);
    }
    fclose( fp);
}

```

参考答案:

```
void SortDat()
```

```
{
```

```
int i,j;
```

PRO swap;

for(i=0;i<MAX-1;i++)

for(j=i+1;j<MAX;j++)

if(strcmp(sell[i].mc,sell[j].mc)<0||

strcmp(sell[i].mc,sell[j].mc)==0&&sell[i].je<sell[j].je)

{ swap=sell[i];sell[i]=sell[j];sell[j]=swap;}

}

五、素数

下列程序的功能是:将大于整数 m 且紧靠 m 的 k 个素数存入数组 xx 中。请编写函数 $\text{num}(\text{int } m, \text{int } k, \text{int } xx[])$ 实现程序的要求,最后调用函数 $\text{readwriteDat}()$ 把结果输出到文件 out.dat 中。

例如:若输入 17,5,则应输出:19 23 29 31 37。

请勿改动主函数 $\text{main}()$ 和输出数据函数 $\text{writeDat}()$ 的内容。

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
void readwriteDAT();
```

```
int isP(int m)
```

```
{
```

```
int i;
```

```
for(i=2;i<m;i++)
```

```
if(m%i==0)return 0;
```

```
return 1;
```

```
}
```

```
void num(int m,int k,int xx[])
```

```
{
```

```
int i=0;
```

```
while(i<k)
```

```
{
```

```
if(isP(m+i))
```

```
xx[i]=m+i;
```

```
i++;
```

```
}
```

```
}
```

```
void readwriteDAT()
```

```
{
```

```
printf("\nPlease enter two integers:");
```

```
scanf("%d,%d",&m,&n);
```

```
num(m,n,xx);
```

```
for(m=0;m<n;m++)
```

```
printf("%d",xx[m]);
```

```
printf("\n");
```

```
readwriteDAT();
```

```
}
```

```
void readwriteDAT()
```

```

        PRO swap;
int m,n,xx[1000], i;
FILE *rf, *wf;
rf = fopen("in. dat", "r");
wf = fopen("out. dat", "w");
for(i=0; i<10; i++) {
    fscanf(rf, "%d%d", &m, &n);
    num(m, n, xx);
    for(m=0; m<n; m++) fprintf(wf, "%d", xx[m]);
    fprintf(wf, "\n");
}
fclose(rf);
fclose(wf);
}

```

注意:求素数的题可以先定义求素数的自定义函数,然后在要求编写的程序中直接调用该函数,既简单又不容易出错,当然源程序如果给了求素数的函数,可以直接调用。

参考答案:

```

void num(int m,int k,int xx[])
{
    int s=0;
    for(m=m+1; k>0; m++)
        if(isP(m)) { xx[s++] = m; k--; }
}

```

六、数字排序

在文件 in. dat 中有 200 组数据,每组有 3 个数,每个数均是三位数。函数 readDat() 读取这 200 组数据存放到结构数组 aa 中。请编制函数 jsSort(), 功能是在 200 组数据中找出条件为每组中的第一个数大于第二个数加第三个数之和,其中满足条件的个数作为函数 jsSort() 的返回值,同时把满足条件的数据存入结构数组 bb 中,再对 bb 中的数据按照每组数据的第一个数加第三个之和的大小按升序排列(第一个数加第三个数的和均不相等),排序后的结果仍重新存入结构数组 bb 中,最后调用函数 writeDat() 把结果 bb 输出到文件 out. dat 中。

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>
typedef struct {
    int x1, x2, x3;
} data;
data aa[200], bb[200];
int jsSort()

```

```

void main()
{
    int count;
    readDat();
    count = jsSort(); /* 返回满足条件的个数 */
    writeDat(count);
}

```

```

readDat()
{
    FILE * in;
    int i;
    in = fopen("in. dat", "r");
    for(i=0; i<200; i++)
        fscanf(in, "%d,%d,%d", &aa[i]. x1, &aa[i]. x2, &aa[i]. x3);
    fclose(in);
}

writeDat(int count)
{
    FILE * out;
    int i;
    out = fopen("out. dat", "w");
    for(i=0; i<count; i++)
        printf("%d,%d,%d 第一个数 + 第三个数 = %d\n", bb[i]. x1, bb[i]. x2,
            bb[i]. x3, bb[i]. x1 + bb[i]. x3);
    fprintf(out, "%d,%d,%d\n", bb[i]. x1,
        bb[i]. x2, bb[i]. x3);
    fclose(out);
}

```

参考答案:

```

int jsSort()
{
    int i, j, k = 0;
    data swap; /* 定义一个结构体变量, 作为交换时的临时存放地 */
    for(i=0; i<200; i++)
        if(aa[i]. x1 > (aa[i]. x2 + aa[i]. x3))
            bb[k++] = aa[i];
}

```

```

/* 先将符合第一个数大于第二个数加第三个数之和的数存入 bb 数组中 */
for(i=0;i<k-1;i++)
    for(j=i+1;j<k;j++)
        if((bb[i].x1+bb[i].x3)>(bb[j].x1+bb[j].x3))
            {swap=bb[i];bb[i]=bb[j];bb[j]=swap;} /* 在 bb 数组中进行排
            序(从小到大) */
return k;
}
/* 将个位数据返回 */

```

七、其他数学计算

请编制函数 ReadDat() 实现从文件 in.dat 中读取 1 000 个十进制整数到数组 xx 中;再编制函数 Compute() 分别计算出 xx 中奇数的个数 odd、偶数的个数 even、平均值 aver 以及方差 totfc 的值,最后调用函数 Writedat() 把结果输出到 oit.dat 文件中。

计算方差的公式如下:

$$\text{totfc} = 1/N \sum_{i=1}^N (\text{xx}[i] - \text{aver})^2$$

原始数据文件存放的格式是:每行存放 10 个数,并用逗号隔开(每个数均大于 0 且小于等于 2 000)。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1000
int xx[MAX], odd=0, even=0;
double aver=0.0, totfc=0.0;
void WriteDat(void);
int ReadDat(void);
void Compute(void)
{
    int i;
    for(i=0;i<MAX;i++)
        xx[i]=0;
    if(ReadDat())
    {
        printf("Can't open the data file in. dat!\n");
        return;
    }
    Compute();
}

```