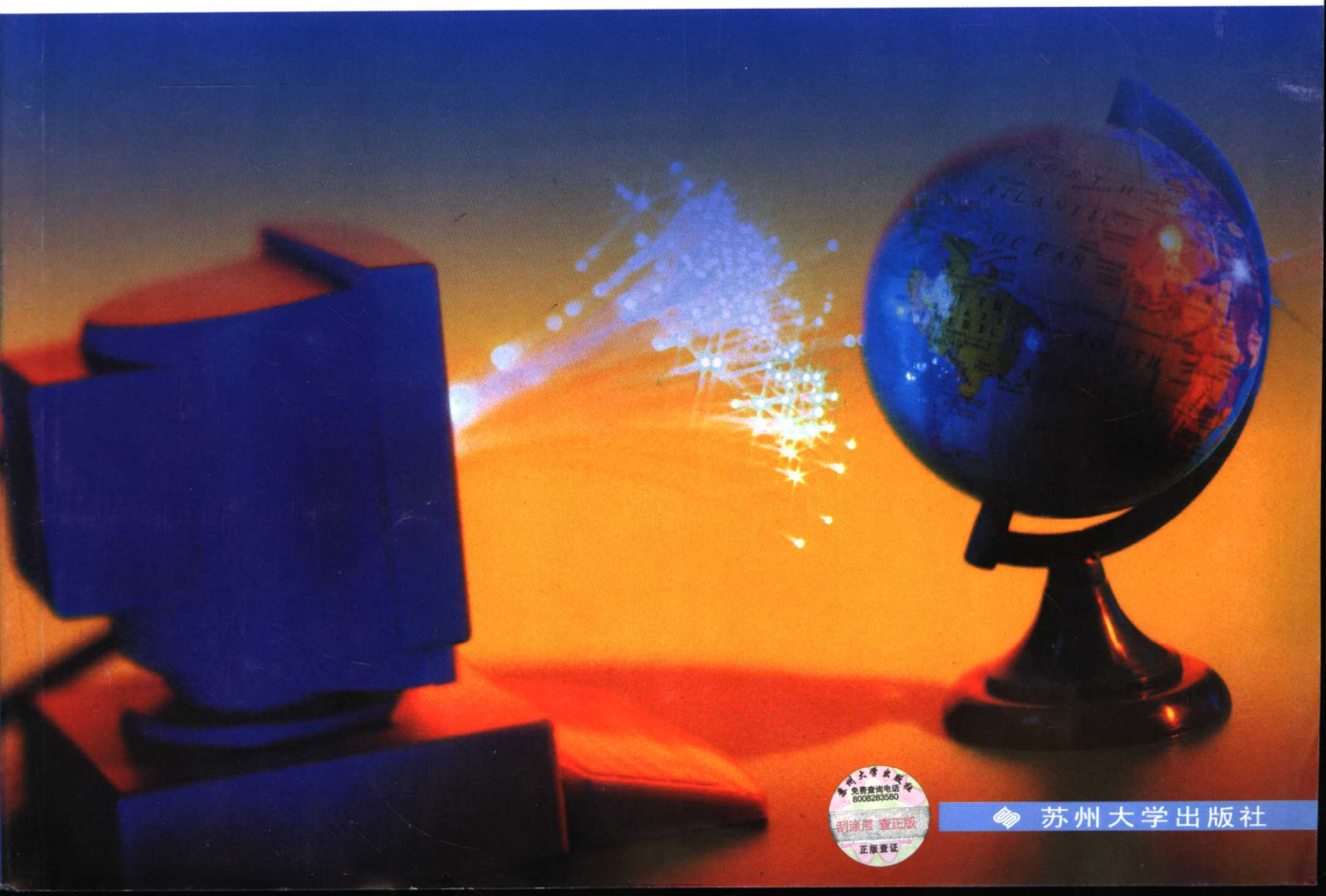


江苏省计算机等级考试辅导用书

# 计算机上机考试 题型归纳与解析

## 二级 C语言



苏州大学出版社

江苏省计算机等级考试辅导用书

# 计算机上机考试 题型归纳与解析

(二级 C 语言)

主 编 骆 健 葛武滇

主 审 王国全

苏州大学出版社

## 图书在版编目(CIP)数据

计算机上机考试题型归纳与解析. 二级 C 语言/骆健,  
葛武滇主编. —苏州: 苏州大学出版社, 2006. 8  
江苏省计算机等级考试辅导用书  
ISBN 7-81090-727-1

I. 计… II. ①骆…②葛… 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(2006)第 095840 号

## 计算机上机考试题型归纳与解析 (二级 C 语言)

骆 健 葛武滇 主编

责任编辑 周建兰

---

苏州大学出版社出版发行  
(地址: 苏州市干将东路 200 号 邮编: 215021)  
常熟高专印刷有限公司印装  
(地址: 常熟市元和路 98 号 邮编: 215500)

---

开本 787mm×1092mm 1/16 印张 41.75(共五册) 字数 1020 千  
2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷  
ISBN 7-81090-727-1/TP·48 定价: 65.00 元  
(共五册)

---

苏州大学版图书若有印装错误, 本社负责调换  
苏州大学出版社营销部 电话: 0512-67258835

# 《计算机上机考试题型归纳与解析》

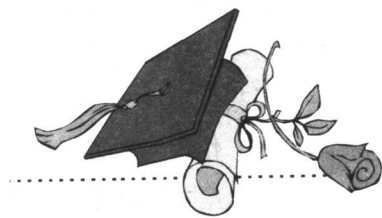
## 编 委 会

编 委 (按姓氏笔画为序)

|     |     |     |     |
|-----|-----|-----|-----|
| 王宏华 | 王国全 | 王景玉 | 尹 静 |
| 朱贵喜 | 严云洋 | 杨章静 | 李千目 |
| 李勇智 | 李筱松 | 吴 婷 | 时兆武 |
| 张居晓 | 陈 智 | 周 松 | 赵 明 |
| 姚昌顺 | 骆 健 | 黄庆宏 | 葛武滇 |

本册主编 骆 健 葛武滇

主 审 王国全



# 前 言

“江苏省计算机等级考试”是面向省内高校非计算机专业学生的计算机水平考试,经过多年的发展,已形成了较大规模,具有较大的影响力。

考试分一级、二级、三级 3 个级别,一级为上机考试,二级分笔试与上机两部分,三级为笔试,笔试和上机考试均要求达到一定标准后考试才算通过。从历年的考试情况来看,上机考试的通过率总体不如笔试,从考生的应试心理分析,由于多数考生平时上机实践的机会较少,训练强度不够,因此更畏惧上机考试。针对这些情况,我们经过精心策划,组织了一批高校计算机基础教学的一线教师,通过对考试大纲和历年考题的深入调研,归纳了上机考试各类试题题型并加以详细解析,编写了这套《计算机上机考试题型归纳与解析》,旨在帮助广大考生进行针对性的考前集训,强化训练,顺利过关。

本套书的几大特点是:

**题型归纳全面**,即较全面地归纳出历年的常考题型,并配以典型例题解析,通过“考点点拨”、“理论链接”等特色栏目,让读者熟悉、理解和掌握各类题型,做到心中有数。

**试题解析详尽**,精心设计了多套上机模拟试题供考生考前集训,所有试题均给出了详细的解答,以便于考生自学自测,做到一书在手,考试无忧。

**定位准确、应试性强**,在摸清考生应试心理的情况下,通过全面归纳题型来揭示命题规律与解题技巧,提供相当数量的实战训练和备考导航,从而突出针对性和实战性。

本套书共 5 册:一级信息技术、二级 Visual Basic、二级 Visual FoxPro、二级 C 语言、二级 Visual C++。作者均为省内多所高校长期从事相关课程教学、考试辅导的一线教师。

本套书以广大应试考生为主要读者对象,同时适于相关课程学习的各类读者参考使用。书中部分原型电子素材可到苏州大学出版社网站([www.sudapress.com](http://www.sudapress.com))查询。

由于时间和水平的原因,书中可能仍有不当之处,欢迎读者批评指正。

《计算机上机考试题型归纳与解析》编委会

## 目 录

## 第一篇 常考题型归纳分析

|                          |      |
|--------------------------|------|
| 一、数值类 .....              | (1)  |
| 题型 1: 素数问题 ★★ .....      | (1)  |
| 题型 2: 整数拆分 ★★ .....      | (4)  |
| 题型 3: 排序问题 ★★ .....      | (10) |
| 题型 4: 其他数学问题 ★★ .....    | (16) |
| 二、字符串类 .....             | (28) |
| 题型 1: 字符串与整数问题 ★★ .....  | (28) |
| 题型 2: 字符串的子串问题 ★★ .....  | (31) |
| 题型 3: 出现次数问题 ★★ .....    | (34) |
| 题型 4: 插入、删除字符问题 ★★ ..... | (38) |
| 题型 5: 其他问题 ★★ .....      | (41) |
| 三、二维数组类 .....            | (48) |
| 题型 1: 最值问题 ★★ .....      | (48) |
| 题型 2: 矩阵运算 ★ .....       | (51) |
| 题型 3: 对角线问题 ★ .....      | (53) |
| 题型 4: 行列变化问题 ★ .....     | (56) |
| 题型 5: 其他问题 .....         | (59) |
| 四、结构体类 .....             | (62) |
| 题型 1: 学生成绩问题 .....       | (62) |
| 题型 2: 其他问题 .....         | (68) |

## 第二篇 模拟试卷精选精解

|               |      |
|---------------|------|
| 上机模拟试卷一 ..... | (73) |
| 上机模拟试卷二 ..... | (76) |
| 上机模拟试卷三 ..... | (79) |
| 上机模拟试卷四 ..... | (82) |
| 上机模拟试卷五 ..... | (85) |
| 上机模拟试卷六 ..... | (88) |
| 上机模拟试卷七 ..... | (91) |







```
20         if(a[i] == 0)printf("%-4d",a[i]);
21         |
```

**【要求】**

1. 将上述程序录入到文件 myf1.c 中,根据题目要求及程序中语句之间的逻辑关系对程序中的错误进行修改。
2. 改错时,可以修改语句中的一部分内容,调整语句次序,增加少量的变量说明或编译预处理命令,但不能增加其他语句,也不能删去整条语句。
3. 改正后的源程序(文件名 myf1.c)必须放在考试软盘的根目录下,供阅卷用,否则不予评分。

**【参考答案】**

1. 第 5 行 int i, j, a[i+1];           改为    int i, j, a[N + 1];
2. 第 15 行 i=i+1;                   改为    j=j+1;
3. 第 20 行 if(a[i] == 0)           改为    if(a[i] != 0)

分析:素数是指除 1 和它本身之外不能被其他任何整数整除的正整数。

筛选法求素数的算法思想通俗地来说就是:在 2~100 间的数中,把不是素数的都划掉,剩下的便都是素数了。

算法的步骤是:先保留 2,然后划掉 2 的所有倍数 4、6 等;保留 3,再划掉所有 3 的倍数 6、9 等;保留 5,再划掉 5 的所有倍数 10、15。

在本程序中,将所有待判定的数放入一维数组中,数组元素的值与下标值相对应。下标为 0 的位置不使用,数组长度应是 100+1 个,即 a[N+1]。

程序的第二个 for 循环所完成的工作是判断当前的 a[i] 是否为素数,如果是,则在第二层的 while 循环中,把 a[i] 的倍数都划掉,即把 a[i] 的倍数标注为 0,表示它不是素数。循环结束后,将所有素数输出,即数组中所有的非 0 值输出。

**典型题 2(编程题):**

1. 编写函数 int prime(int n),判断 n 是否为素数,若是素数,函数返回 1;否则,函数返回 0。
2. 编写 main 函数,从键盘输入查找素数的范围,调用 prime 函数,将此范围内的所有素数按从大到小的顺序排序,并写入文件 myf2.out 中。

例如,查找范围为 1~10,输出结果应为:7 5 3 2。

**【要求】**

1. 将源文件取名为 myf2.c,输出结果文件取名为 myf2.out。
2. 数据文件的打开、使用和关闭等操作均用 C 标准库中缓冲文件系统的文件操作函数实现。
3. 源程序文件和运行结果文件均须保存在考试软盘的根目录下,供阅卷用。
4. 不要将 myf2.obj、myf2.exe 保存到考试软盘中。

**【参考答案】**

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

```

int prime( int n)
{
    int m;
    for(m = 2; m <= sqrt( n); m ++ )
        if( n% m == 0) return 0;
    return 1;
}

main()
{
    int m, X, Y;
    FILE * out;
    if( ( out = fopen( "myf2. out", "w" ) ) == NULL)
    {
        printf( "open file myf2. out failed! \n" );
        exit( 0 );
    }

    printf( "input X, Y: " );
    scanf( "%d, %d", &X, &Y );
    for( m = Y; m > X; m -- )
    {
        if( prime( m ) )
        {
            fprintf( out, "%d ", m );
        }
    }

    fclose( out );
}

```

**分析：**测试一个数  $n$  是否为素数的基本思想是：用  $2 \sim \sqrt{n}$  之间的数依次去除  $n$ ，只要有一个数能将  $n$  整除， $n$  就不是素数。

算法步骤如下[设除数为  $m$ ， $m$  的值由 2 变化到  $\sqrt{n}$ ]：

- ① 使  $n$  除以  $m$ ，得到余数。
- ② 判断余数是否等于 0，如果余数等于 0，表示  $n$  能被  $m$  整除， $n$  不是素数，算法结束；如果余数不等于 0， $m$  的值加 1。
- ③ 判断  $m \leq \sqrt{n}$  是否成立，是则返回重新执行①；否则，确定  $n$  是素数，算法结束。

**理论链接：**

1. 使用函数  $\text{sqrt}()$ 。

使用  $\text{sqrt}()$  等数学函数时，应在源文件中使用“ $\#include < \text{math. h} >$ ”。

函数原型： $\text{double sqrt}(\text{double } x)$ ；

功能：计算  $x$  的平方根。

2. 如何将查找到的素数按从大到小的顺序输出？

在  $\text{main}$  函数的  $\text{for}$  循环中，以  $Y$  的值（较大值）定为循环的起点，以  $X$  的值（较小值）定



**【参考答案】**

1. 第 8 行  $b = a[i] \% 100$ ;            改为      $b = a[i] \% 1000 / 100$ ;  
 2. 第 10 行  $e = a[i] / 10$ ;            改为      $e = a[i] \% 10$ ;  
 3. 第 12 行  $\text{if}((d - b + x) * e = 4)$  改为      $\text{if}((d - b + x) * e == 4)$

分析: 该题的关键在于如何取出一个数的个、十、百、千位上的数。

$a[i] \% 10$  能取得  $a[i]$  个位上的数;

$a[i] \% 100 / 10$  可求得十位上的数, 即先通过  $a[i] \% 100$  求出  $a[i]$  的后两位数, 然后再将后两位数除 10, 求得的商即是  $a[i]$  十位上的数;

$a[i] \% 1000 / 100$  可求得百位上的数, 即先通过  $a[i] \% 1000$  求出  $a[i]$  的后三位数, 然后再将后三位除 100, 求得的商即是  $a[i]$  百位上的数。

$a[i] / 1000$  能取得  $a[i]$  千位上的数。

**理论链接:**

1. 以上分析中所得出的整数拆分方法可应用到 2 位、3 位或 5 位的整数中。以 5 位整数为例, 一个 5 位整数  $X$  可表示为:  $a * 10000 + b * 1000 + c * 100 + d * 10 + e$ , 则拆分的方法如下:

$$a = X / 10000$$

$$b = X \% 10000 / 1000$$

$$c = X \% 1000 / 100$$

$$d = X \% 100 / 10$$

$$e = X \% 10$$

2. 特别注意赋值号“=”与等于号“==”的区别。

程序中, 表示式  $\text{if}((d - b + x) * e = 4)$  的原意是要判断个、十、百、千位上的数运算后的值是否等于 4, 但“=”是赋值号, 不能用来进行比较运算。这是考生在改错题和编程题中常出现的错误。

**典型题 2 (编程题):**

1. 编写函数  $\text{void fun}(\text{int } a, \text{int } b, \text{int } *c, \text{int } *d)$ , 函数 fun 的功能是: 将两个两位数的正整数  $a$ 、 $b$  合并形成两个新整数, 存放在  $c$  和  $d$  中。

合并的方式如下:

将  $a$  数的十位和  $b$  数的十位依次放在  $c$  数的千位和百位上, 将  $a$  数的个位和  $b$  数的个位依次放在  $c$  数的十位和个位上。

将  $a$  数的十位和  $b$  数的个位依次放在  $d$  数的千位和百位上, 将  $a$  数的个位和  $b$  数的十位依次放在  $d$  数的十位和个位上。

2. 编写 main 函数, 从键盘输入测试数据, 调用 fun 函数, 将得到的两个新整数写入文件 myf2.out 中。

例如, 当  $a = 45, b = 12$ 。调用该函数后,  $c = 4152, d = 4251$ 。

**【要求】**

1. 将源文件取名为 myf2.c, 输出结果文件取名为 myf2.out。
2. 数据文件的打开、使用和关闭等操作均用 C 标准库中缓冲文件系统的文件操作函数

实现。

- 源程序文件和运行结果文件均须保存在考试软盘的根目录下,供阅卷用。
- 不要将 myf2.obj、myf2.exe 保存到考试软盘中。

**【参考答案】**

```
#include <stdio.h>
void fun(int a,int b,long *c,long *d)
{
    *c = a/10 * 1000 + b/10 * 100 + a%10 * 10 + b%10;
    *d = a/10 * 1000 + b%10 * 100 + a%10 * 10 + b/10;
    printf("%ld,%ld", *c, *d);
}
main()
{
    int a,b;
    long t,r;
    FILE *out;
    if( (out = fopen("myf2.out","w")) == NULL)
    {
        printf("open file myf2.out failed! \n");
        exit(0);
    }
    printf("\nPlease enter a,b:");
    scanf("%d,%d", &a,&b);
    fun(a,b, &t,&r);
    fprintf(out,"%ld\n%ld\n",t,r);
    fclose(out);
}
```

**分析:** 本题是将上题中的整数拆分方法应用到两位整数上。 $a/10$  可得到  $a$  数十位上的数,再乘以 1000,此数将放在  $c$  数的千位上。 $b/10$  可得到  $b$  数十位上的数,乘以 100,将此数放在  $c$  数的百位上。 $a\%10$  可得到  $a$  数的个位上的数,再乘以 10,将此数放在  $c$  数的十位上。 $b\%10$  可得到  $b$  数的个位上的数,将此数放在  $c$  数的个位上。将以上 4 个数值相加,即得到所求的  $c$  数。同理可得  $d$  数。

**理论链接:**

如何将函数中的多个计算结果返回到主调函数中?

函数通常情况下只能返回一个值,当函数中有多个值需要返回时,可考虑通过指针类型或数组类型的形参返回。

**典型题 3(改错题):**

函数 long fun(long s)的功能是:将长整型数中每一位上为偶数的数依次取出组成一个新数,并将此数通过函数返回。

例如,当  $s$  中的数为 876214 时,返回的数为 8624。









的值右移一位。

例如, num 的值为 252 时, “k \* = num% 10;” 语句取得变量 num 个位上的值, 并累乘到变量 k 中, 然后 num 右移一位, 其值变为 25。再次执行循环时, “k \* = num% 10;” 语句的再次取得变量 num(此时其值为 25) 个位上的值(实质是原来 252 的十位上的 5) 并乘到变量 k 中。这样从低位向高位把 num 中各位上的数逐步取出, 累乘到变量 k 中。

为了使 num 的值右移一位, 应使 num 除以 10, 即 “num / = 10;”, 而不是 “num \ = 10;”。最后, 需要将各位数值之积 k 返回。

### 题型 3: 排序问题 ★★

#### 考点点拨:

解决排序问题的算法有很多。例如, 选择排序、插入排序、快速排序等都是经常考查的知识点。

#### 典型题 1(改错题):

函数 void Merge(int A[], int B[], int Temp[], int len\_a, int len\_b) 的功能是: 将两个升序序列 a 和 b 合并产生一个新的升序序列 temp。

含有错误的源程序如下:

```

1      #include <stdio. h >
2      void Merge(int A[], int B[], int Temp[], int len_a, int len_b)
3      {
4          int i = 0, j = 0, k;
5          while(( i < len_a) && ( j < len_b))
6              {
7                  if( A[ i ] <= B[ j ]) Temp[ k ++ ] = A[ i ++ ];
8                  Temp[ k ++ ] = B[ j ++ ];
9              }
10         while( i > len_a) Temp[ k ++ ] = A[ i ++ ];
11         while( j < len_b) Temp[ k ++ ] = B[ j ++ ];
12     }
13     main()
14     {
15         int a[] = { 8, 12, 45, 78 }, b[] = { 1, 23, 28, 79, 90 }, temp[ 9 ];
16         int alen = 4, blen = 5, i;
17         Merge( a, b, temp, alen, blen );
18         for( i = 0; i < alen + blen; i ++ )
19             printf( "%4d", temp[ i ] );
20     }
    
```