

## 上机考试策略与技巧

三级 C 语言程序设计的上机考试一般可以在考试前的一个月左右开始复习，如果以前没有接触过 C 语言，则需要更长的时间来准备，大约在考前 2~3 个月开始复习即可。考生在准备上机考试的时候，应该首先将 C 语言的基础知识复习一遍，重点为：

- 循环（包括 for 循环和 while 循环），分支（主要为 if 语句）和函数。
- 结构体数组和字符串。
- 细节问题，例如：“=”与“==”的区别；“break”与“continue”的区别；“|”与“||”、“&”与“&&”的区别；“<<”与“>>”的使用等。
- 另外，在准备考试的时候一定要多上机实践，熟悉 TC 2.0 环境下的操作。例如，Ctrl+F9 为一次性执行编译、连接和运行操作，Alt+F5 为显示结果，F2 为存盘等。

考点一般都是让考生抽签决定自己的机器号，待考生输入准考证号后，系统随机抽出一份试题，在考试的过程中，考生一定要在指定的目录，即系统默认目录下答题，并保存。一旦不小心退出了系统，应该及时返回，或者向监考老师举手示意。

三级 C 语言程序设计的上机考试只有一道编程题，开始考试后，考生不应该急于编程，而应先仔细分析问题，待有思路后再开始动手编程。在编程完成后按下 Ctrl+F9 运行程序，若编译没有通过，修改错误时，应先从第一个错误开始，因为很多时候后面的错误是由前面的错误引起的，注意不要改动程序中原来已经给出的读写函数或主函数。

考生在完成题目后，不应急于离开考场，检查几遍也是非常有必要的，检查后若有修改，仍要存盘。

程序设计的算法并不惟一，对于同一个题目，会有多个不同的解法，考生如果对自己的想法胸有成竹，不妨大胆进行尝试，不必拘泥于课本和参考书上的答案。

在考试时，考生可以将草稿纸和笔带进考场，当思路不是很清晰的时候，在纸上验算一下，可以起到事半功倍的效果，考生必须在规定时间内完成考试，否则系统将强行终止考试，此时只有监考人员输入延时密码后方可继续操作，所以考生一定要注意考试时间，考试时要精力集中，不要浪费时间。

在三级的上机考试中，编程题大概可以分为以下五类。

- 数据分解类型，即把数字各个位上的数字分离出来，再进行相应的处理（排序和筛选等）。
- 结构体数据类型，此类题目多要求处理一个产品的销售记录。
- 对字符串的处理类型，多为对一篇英文文章的各个字符进行相应的处理。
- 按条件查找类型，即题目中给出查找条件在指定的数据范围内（多为自然数范围）进行查找。
- 实际应用类型，此类题目多数和现实生活有关（例如统计选票等）。

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

### 第 1 题

请编写一个函数 jsValue(int m,int k,int xx[])，该函数的功能是：将大于整数 m 且紧靠 m 的 k 个素数存入数组 xx 传回。

最后调用函数 `writeDat()` 读取 10 组数据，分别得出结果且把结果输出到文件 `out.dat` 中。部分程序已经给出。

例如：若输入 17, 5, 则应输出：19, 23, 29, 31, 37。

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();
void num(int m,int k,int xx[])
{
}
main()
{
    int m,n,xx[1000];
    clrscr();
    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()
{
    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);
}
```

}

**【答案】**

```

void num(int m,int k,int xx[])
{
    int data=m+1;
    int half,n=0,i;
    while(1)
    {
        half=data/2;
        for(i=2;i<=half;i++)
            if(data%i==0)
                break;
        if(i>half)
        {
            xx[n]=data;
            n++;
        }
        if(n>=k)
            break;
        data++;
    }
}

```

**【解析】**

该程序属于按条件查找类型的题目，考核的知识点主要为：

- (1) 判断一个数字是否为素数。
- (2) 统计素数的个数。

本题的解题思路为：从  $m+1$  开始逐个数字判断是否为素数（即只能被 1 和它本身整除的整数）。若是素数则存入数组  $xx$  中，一直寻找连续的  $k$  个素数为止。判断是否为素数，可以通过除以 2 到它的一半之间所有的数，比如，判断 17 是否为素数，则需除以 2、3、4、5、6、7、8 若都不能整除，则该数字就一定是素数。寻找连续的  $k$  个素数，可以设置一个记录个数的变量，每找到一个素数时该记录变量加 1。

程序的流程为：首先从键盘输入两个十进制整数，以第一个整数作为基础开始寻找素数，第 2 个整数作为需要寻找素数的个数，然后把  $num()$  函数的处理结果，通过  $readwriteDat()$  函数写回文件  $out.dat$  中。

在  $num()$  函数中，使用一个  $while$  循环查找  $n$  个素数，每次查找成功后  $n$  值将加 1。因为不能确定循环执行的具体次数，所以设定  $while$  循环的条件恒为真（条件为“1”），并加入跳出循环的控制语句，即当  $n \geq k$ （超过所需的素数个数）时循环结束。

进入  $while$  循环体后，首先使用  $for$  循环，确定当前数字是否为素数，即判断当前数字  $data$  是否能被 2 到它一半之间的任意数整除，如果能被整除，则说明此数不是素数，程序

通过 `break` 语句跳出 `for` 循环（此时  $i \leq \text{half}$ ），否则循环执行到  $i > \text{half}$  时，程序正常退出 `for` 循环体，说明当前数字 `data` 是素数，这里的 `i` 值既作为 `for` 循环计数器，又作为当前数字是否为素数的标志。

`if` 语句将根据此时 `i` 值的大小判断 `data` 是否为素数，如果是素数，则将 `data` 存入数组 `xx` 中，同时记录素数个数的变量 `n` 加 1，接着判断 `n` 的值是否达到了题目要求的个数，如果达到了，则跳出 `while` 循环，`num()` 函数结束，否则 `data` 加 1 后重新进入 `while` 循环体。直到找到满足个数的素数，`num()` 函数结束。

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

## 第 2 题

已知数据文件 `IN.DAT` 中存有 200 个四位数，并已调用读函数 `readDat()` 把这些数存入数组 `a` 中，请考生编制一函数 `jsVal()`，其功能是：如果四位数个位上的数字均是 0 或 2 或 4 或 6 或 8，则统计出满足此条件的个数 `cnt`，并把这些四位数按从大到小的顺序存入数组 `b` 中。最后 `main()` 函数调用写函数 `writeDat()` 把结果 `cnt` 以及数组 `b` 中符合条件的四位数输出到 `OUT.DAT` 文件中。

注意：部分程序已经给出。

程序中已定义数组：`a[200]`，`b[200]`，已定义变量：`cnt`。

请勿改动数据文件 `IN.DAT` 中的任何数据、主函数 `main()`、读函数 `readDat()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>
#define MAX 200
int a[MAX], b[MAX], cnt = 0;
void jsVal()
{
    .
    .
    .
}
void readDat()
{
    .
    .
    .
    int i;
    FILE *fp;
    fp = fopen("in.dat", "r");
    for(i = 0; i < MAX; i++)
        fscanf(fp, "%d", &a[i]);
    fclose(fp);
}
void main()
{
    .
    .
    .
    int i;
    readDat();
```

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

### 【答案】

```
void jsVal()
{
    int bb[4];
    int i,j,k,flag;
    for(i=0;i<200;i++)
    {
        bb[0]=a[i]/1000;
        bb[1]=a[i]%1000/100;
        bb[2]=a[i]%100/10;
        bb[3]=a[i]%10;
        for(j=0;j<4;j++)
        {
            if(bb[j]%2==0)
                flag=1;
            else
            {
                flag=0;
                break;
            }
        }
        if(flag==1)
```



## 第 3 题

函数 ReadDat()实现从文件 IN.DAT 中读取一篇英文文章存入字符串数组 xx 中；请编制函数 StrOR()，其函数的功能是：以行为单位依次把字符串中所有小写字母 o 左边的字符串内容移到该串的右边存放，然后把小写字母 o 删除，余下的字符串内容移到已处理字符串的左边存放，之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 main()函数调用函数 WriteDat()把结果 xx 输出到文件 OUT5.DAT 中。

例如：原文：n any field.Yu can create an index

you have the correct record.

结果：n any field. Yu can create an index

rd. yu have the crrect rec

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

int ReadDat(void);
void WriteDat(void);
void StrOR(void)
{
}

void main()
{
    clrscr();
    if(ReadDat())
    {
        printf("数据文件 IN.DAT 不能打开!\n\007");
        return;
    }
    StrOR();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
```

```

int i = 0 ;
char *p ;
if((fp = fopen("IN.DAT", "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 ;
    clrscr() ;
    fp = fopen("OUT5.DAT", "w") ;
    for(i = 0 ; i < maxline ; i++)
    {
        printf("%s\n", xx[i]) ;
        fprintf(fp, "%s\n", xx[i]) ;
    }
    fclose(fp) ;
}

```

**【答案】**

```

void StrOR(void)
{
    int i,j,k,index,strlen;
    char ch;
    for(i=0;i<maxline;i++)
    {
        strlen=strlen(xx[i]);
        index=strlen;
        for(j=0;j<strlen;j++)
            if(xx[i][j]=='o')
            {

```



**【解析】**

本题的解题思路是：以行为单位查找并删除字母“o”，然后根据题意进行字符串的互换。实现字符串的互换时，把左边字符串移到右边，与把右边字符串移到左边其结果一样，但是后者相对简单，故本题采用后者。删除字符“o”；可以通过找到每行中的字母“o”后，将“o”后面的字母依次向前移一位，即可以把“o”删除。把字符“o”右边的字符串移到左边，可以将最后一位字符移动到字符串的首部，并将其他的字符依次向后移一位，直到原来字母“o”所在位置上的字符也被移到字符串首部为止，确定原来字符“o”所在的位置，可以设定一个记录字符“o”位置的变量。

在 StrOR()函数中,最外层 for 循环的作用是使下面的操作逐行进行,首先通过库函数 strlen()得到每行字符串的实际长度并赋给 strl,循环到 i=maxline,即访问过所有行时结束,然后进入内嵌的第二层 for 循环,循环到 j=strl,即访问过本行的所有字符时结束,在循环体中,通过 if 语句依次判断当前元素是否为字符“o”,若是“o”则进入第三层循环体,在循环体中将字符“o”后面的元素依次向前移一位,此时字符“o”已经被它后面的元素所取代,然后再把本行最后一个字符置空,此时 j 的值为字符“o”原来所在数组元素的下标,把 j 赋给 index,即用 index 来标记字符“o”原来所在的位置,当 k=strl-1,即“o”后面所有的字符都依次向前移动一位后,本层 for 循环结束。进入下一个 for 循环,在以下程序中,先把数组中每行的最后一个元素赋给 ch 后,进入下一重循环,将从 index 标记的位置开始逐个向后移一位,并且把最后一位移到本行的开头处,一直循环到 index 标记的位置也被移到开头处,函数结束。

\*\*\*\*\*

## 第 4 题

函数 ReadDat() 实现从文件 IN.DAT 中读取一篇英文文章并存入到字符串数组 xx 中，请编制函数 StrOL()，其函数的功能是：以行为单位对行中以空格或标点符号为分隔的所有单词进行倒排。最后把已处理的字符串（应不含标点符号）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 OUT6.DAT 中。

例如：原文：You He Me

I am a student.

结果：Me He You

student a am I

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。

部分程序已经给出。

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 StrOL(void)
{

}

void main()
{
    clrscr();
    if(ReadDat())
    {
        printf("数据文件 IN.DAT 不能打开! \n\007");
        return;
    }
    StrOL();
    WriteDat();
}

int ReadDat(void)
{
    FILE *fp;
```

```

int i=0;
char *p;
if((fp=fopen("IN.DAT","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;
    clrscr();
    fp=fopen("OUT6.DAT","w");
    for(i=0;i<maxline;i++)
    {
        printf("%s\n",xx[i]);
        fprintf(fp,"%s\n",xx[i]);
    }
    fclose(fp);
}

```

### 【答案】

```

void StrOL(void)
{
    int i,j,k,trl,l;char c;
    for(i=0;i<maxline;i++)
        for(j=0;j<strlen(xx[i]);j++)
        {
            c=xx[i][j];
            if ((c>='A'&&c<='Z')||(c>='a'&&c<='z')||c==' ');
            else xx[i][j]=' ';
        }
    for(l=0;l<maxline;l++)

```

```

{
    char ch[80]={0};
    char pp[80]={0};
    strl=strlen(xx[l]);
    i=strl-1;
    k=1;
    while(1)
    {
        while(((xx[l][i]>='a'&&xx[l][i]<='z')||(xx[l][i]>='A'
            &&xx[l][i]<='Z'))&&i>=0)
        {
            for(j=k;j>=0;j--)
                pp[j+1]=pp[j];
            pp[0]=xx[l][i];
            k++;i--;
        }
        strcat(ch,pp);
        strcpy(pp, "");
        k=1;
        if(i== -1)
            break;
        while((xx[l][i]<'A' || xx[l][i]>'z')&&i>=0)
        {
            for(j=k;j>=0;j--)
                pp[j+1]=pp[j];
            pp[0]=xx[l][i];
            k++; i--;
        }
        treat(ch,pp);
        strcpy(pp, "");
        k=0;
        if(i== -1)
            break;
    }
    strcpy(xx[l],ch);
}
}

```

**【解析】**

本程序属于对字符串处理类型的题目，考核的知识点主要为：（1）删除字符串中的标

点符号；(2) 对删除后的字符串重新进行排序。

本题的解题思路是：先找到并删除每行字符串中的非英文字符，再从每行最后一个单词开始，依次将每个单词存入一个临时数组中，然后将临时数组连接起来，结果即实现将所有单词倒排。删除非英文字符时，可以通过将其赋值为空字符来实现，连接临时数组通过字符串连接函数 `strcat()` 来实现。本题在此类题目中属于难度较大的一道，读者应该对程序的核心算法多加分析。

程序的流程是：首先调用 `ReadDat()` 函数，将数据读入到数组 `xx` 中，然后调 `StrOL()` 函数对数据进行处理，最后调用 `WriteDat()` 函数，将结果输出到文件 `OUT6.DAT` 中。

在 `StrOL()` 函数中，首先采用两重 `for` 循环，将数组 `xx[i]` 中不是字母也不是空格的元素全部赋值为空字符，然后进入下面一层 `for` 循环（此处声明了两个字符型数组 `pp` 和 `ch`，用于存放处理后的临时数据），该层 `for` 循环在 `i=maxline`，即访问过所有行时结束。由于不确定循环的次数，所以设定最外层 `while` 循环条件恒为真（非 0 即为真），在 `i=-1` 时通过 `break` 语句跳出循环，程序直接进入内嵌 `while` 循环中，本层 `while` 循环的功能是，从 `xx[i]` 数组，即文章的每一行中最后的一个字符开始向前查找，每找到一个单词就将其存入一个临时数组中，最后按照找到单词的先后顺序将临时数组连接起来，并且各个单词之间用空格分开。

在第二个 `while` 循环中，当 `xx[i]` 数组中的元素为大写字母或小写字母且 `i >= 0` 时进入 `for` 循环，该循环体的功能是将 `pp` 数组中的元素依次向后移一位。然后从 `xx[i]` 数组中的最后一个元素开始，每次 `for` 循环结束后，就将其存入 `pp[0]` 中，然后 `k` 和 `i`（由后向前）都指向下一个元素，重新进入到 `while` 循环当中，本层 `while` 循环一直执行到 `xx[i]` 数组中出现空格时结束。`pp` 数组即为存放每一个单词的临时数组，再将 `pp` 数组的内容存入 `ch` 数组中，`ch` 数组为将各个单词倒排的临时数组。此时需调用字符串连接函数 `strcat()`，之后将 `pp` 数组置空。下一层 `while` 循环的条件为当 `xx[i]` 数组中的元素不是大写字母且 `i >= 0` 时，进入内嵌 `for` 循环，该 `for` 循环的功能依然为将 `pp` 数组中的元素依次向后移一位。然后将 `xx[i]` 数组中的最后一个元素开始，每次 `for` 循环结束后，就将其存入 `pp[0]` 中，然后 `k` 和 `i`（由后向前）都指向下一个元素，重新进入到 `while` 循环体中。本层 `while` 循环结束后，将 `pp` 数组连接在 `ch` 数组后，同时将 `pp` 数组置空，最外层 `while` 循环执行到 `i=-1`，即本行所有元素都访问过后结束，字符串排序结束，最后将 `ch` 数组中的内容拷贝到 `xx[i]` 数组中，`for` 循环执行到 `xx` 数组中的所有元素均被访问过，然后退出循环体，`StrOL()` 函数结束。

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

## 第 5 题

在文件 `in.dat` 中有 200 个正整数，且每个数均在 1000 至 9999 之间。函数 `ReadDat()` 读取这 200 个数存放到数组 `aa` 中。请编制函数 `jsSort()`，其函数的功能是：要求按每个数的后三位的大小进行升序排列，然后取出满足此条件的前 10 个数依次存入数组 `bb` 中，如果后三位的数值相等，则按原先的数值进行降序排列。最后调用函数 `WriteDat()` 把结果 `bb` 输出到文件 `out.dat` 中。

例：处理前 6012 5099 9012 7025 8088

处理后 9012 6012 7025 8088 5099

部分程序已经给出。

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>
int aa[200],bb[10];
void jsSort()
{

}

void main()
{
    ReadDat();
    jsSort();
    WriteDat();
}
ReadDat()
{
    FILE *in;
    int i;
    in=fopen("in.dat","r");
    for(i=0; i<200; i++)
        fscanf(in,"%d",&aa[i]);
    fclose(in);
}
WriteDat()
{
    FILE *out;
    int i;
    clrscr();
    out=fopen("out.dat","w");
    for(i=0; i<10; i++)
    {
        printf(" %d",bb[i]);
        fprintf(out,"%d\n",bb[i]);
    }
    fclose(out);
}
```

**【答案】**

```
void jsSort()
```

```

{
    int i,j,data;
    for(i=0;i<199;i++)
        for(j=i+1;j<200;j++)
            {
                if (aa[i]%1000>aa[j]%1000)
                    {
                        data=aa[i];
                        aa[i]=aa[j];
                        aa[j]=data;
                    }
                else if(aa[i]%1000==aa[j]%1000)
                    if(aa[i]<aa[j])
                        {
                            data=aa[i];
                            aa[i]=aa[j];
                            aa[j]=data;
                        }
            }
}
for(i=0;i<10;i++)
    bb[i]=aa[i];
}

```

## 第 6 题

在文件 in.dat 中有 200 个正整数，且每个数均在 1000 至 9999 之间。函数 ReadDat() 读取这 200 个数存放到数组 aa 中。请编制函数 jsSort()，其函数的功能是：要求按每个数的后三位的大小进行降序排列，然后取出满足此条件的前 10 个数依次存入数组 bb 中，如果后三位的数值相等，则按原先的数值进行升序排列。最后调用函数 WriteDat() 把结果 bb 输出到文件 out.dat 中。

例：处理前 9012 5099 6012 7025 8088

处理后 5099 8088 7025 6012 9012

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>
int aa[200],bb[10];
void jsSort()
{

}

void main()
{
    ReadDat();
    jsSort();
    WriteDat();
    system("pause");
}

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

WriteDat()
{
    FILE *out;
    int i;
    clrscr();
```



```

out=fopen("out.dat","w");
for(i=0; i<10; i++)
{
    printf("i=%d,%d\n",i+1,bb[i]);
    fprintf(out,"%d\n",bb[i]);
}
fclose(out);
}

```

### 【答案】

```

void jsSort()
{
    int i,j,data;
    for(i=0;i<199;i++)
        for(j=i+1;j<199;j++)
        {
            if(aa[i]%1000<aa[j]%1000)
            {
                data=aa[i];
                aa[i]=aa[j];
                aa[j]=data;
            }
            else if(aa[i]%1000==aa[j]%1000)
                if(aa[i]>aa[j])
                {
                    data=aa[i];
                    aa[i]=aa[j];
                    aa[j]=data;
                }
        }
    for(i=0;i<10;i++)
        bb[i]=aa[i];
}

```

### 【解析】

分析过程同第 5 题。注意本题是先按后三位进行降序排列，若后三位相等则按原数字升序排列。

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

### 第 7 题

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