



第 1 部分 历届上机试题汇编

1. 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 `readwriteDat()` 读取 10 组数据, 分别得出结果且把结果输出到文件 `out1.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 jsValue (int m,int k,int xx[])
{
}

main()
{
    int m,n,xx[1000];
    clrscr();
    printf(" \nPlease enter two integers:");
    scanf("%d,%d",&m,&n);
    jsValue (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("in1.dat","r");
    wf=fopen("out1.dat","w");
    for(i=0;i<10;i++)
    {
        fscanf(rf,"%d %d",&m,&n);
        jsValue (m,n,xx);
        for(m=0;m<n;m++)
            fprintf(wf,"%d",xx[m]);
        fprintf(wf,"\n");
    }
    fclose(rf);
    fclose(wf);
}
```

【答案】

```
void jsValue (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】

考点：(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 $jsValue()$ 函数的处理结果通过 $readwriteDat()$ 函数写回文件 $out1.dat$ 中。

在 $jsValue()$ 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 $i \leq half$ ），否则循环执行到 $i > half$ 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，又作为是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， $jsValue()$ 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， $jsValue()$ 函数结束。

2. 在文件 $in2.dat$ 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 个数字存放到数组 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 $out2.dat$ 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("in2.dat","r");
    for(i=0;i<200;i++)
        fscanf(in,"%d",&aa[i]);
    fclose(in);
}

WriteDat()
{
    FILE *out;
    int i;
    clrscr();
    out = fopen("out2.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];
}

```

【解析】

考点：(1)数据的分解；

(2)分解后的筛选；

(3)筛选后的排序。

本题的解题思路是：对每个四位数的后三位数字进行比较排序，若后三位相等则按原始值进行排序。可以通过“%”来取出当前数的后三位数字，比较排序的算法是每次选定一个位置的元素和它后面的所有元素相比较，如果比后面的元素大则两者交换，若相等则按实际大小排序，比较一趟完成后，此位置上元素的后三位数字，都不大于其后面所有元素的后三位数字。比较一直进行到所有元素均被访问后结束，一共需要比较 199 趟，每趟需要比较 199 - j 次。

程序的流程是：首先调用了 ReadDat() 函数读入数据并且存放在数组 aa 中，然后调用 jsSort() 函数对数据进行操作，最后由 WriteDat() 函数将结果写回文件 out2.dat 中去。

在 jsSort() 函数中，前面两重 for 循环的功能是对数组 aa 中的元素进行排序，其排序的过程是根据每个元素的后三位数字大小进行升序排列。其中 aa[i] % 1000 操作是舍去千位上的数字，得到百位、十位和个位数字，然后通过下面 else if 语句和内嵌 if 语句进行判断，如果元素的后三位相等，则按实际大小进行降序排列。最后执行一个 for 循环，将排序好的前十位数字存入数组 bb 中，jsSort() 函数结束。

3. 已知数据文件 in3.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：依次从数组 a 中取出一个四位数，如果该四位数连续大于该四位数以后的 5 个数且该数是奇数，则统计出满足此条件的个数 cnt，并把这些四位数按从小到大的顺序存入数组 b 中，最后调用写函数 writeDat() 把结果 cnt 以及数组 b 中符合条件的四位数输出到 out3.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 MAX200
int a[ MAX ], b[ MAX ], cnt = 0;
void jsVal()
{
}

void readDat()
{
    int i;

```

```

FILE *fp;
fp = fopen( "in3. dat", "r" );
for( i = 0; i < MAX; i ++ ) fscanf( fp, "%d", &a[ i ] );
fclose( fp );
}

void main( )
{
    int i;
    readDat( );
    jsVal( );
    printf( " Satisfies the condition number: %d\n", cnt );
    for( i = 0; i < cnt; i ++ ) printf( " %d ", b[ i ] );
    printf( "\n" );
    writeDat( );
}

writeDat( )
{
    FILE *fp;
    int i;
    fp = fopen( "out3. 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 i, j, flag = 0;
    int k;
    for( i = 0; i < MAX - 5; i ++ )
    {
        for( j = i + 1; j <= i + 5; j ++ )
        {
            if( a[ i ] > a[ j ] ) flag = 1;
            else { flag = 0; break; }
            if( a[ i ] % 2 == 0 )
                flag = 0;
        }
        if( flag == 1 )
            { b[ cnt ] = a[ i ]; cnt ++; }
    }

    for( i = 0; i < cnt - 1; i ++ )
        for( j = i + 1; j < cnt; j ++ )
            if( b[ i ] > b[ j ] )
                { k = b[ i ]; b[ i ] = b[ j ]; b[ j ] = k; }
}

```

【解析】

考点:(1)按条件筛选;

(2)统计满足条件的个数;

(3)排序算法。

本题的解题思路是:对数组 **a** 中的元素逐个进行判断,若其能连续大于后面的 5 个数(此处可以通过设置一个标志变量来标识 **a[i]** 能否连续大于其后面的 5 个数字)且为奇数时 把它存入 **b** 数组中 并对数组 **b** 进行排序。此排序算法为每次选定一个位置的元素和它后面的所有元素相比较,如果比后面的元素大,则两者交换,完成比较一趟后,此位置的元素就比其后面所有的元素都小。比较一直进行到所有元素均被访问后结束,一共需要比较 **cnt** 趟 每趟需要比较 **cnt-j** 次。

程序的流程是 先调用 **readDat()** 函数 读入数据到数组 **a** 中 然后调用 **jsVal()** 函数 对数据进行处理再由 **writeDat()** 函数 写回文件 **out3.dat** 中去。

在 **jsVal()** 函数中 通过双重 **for** 循环和内嵌 **if** 语句对数组 **a** 中的元素逐个判断,每当条件满足时就将标记变量 **flag** 设置为 1 否则将 **flag** 设置为 0。外层 **for** 循环结束的条件是 **i=MAX-5** (即一直比较到数组当中倒数第 6 个元素)在 **for** 循环中,当 **a[i]** 连续大于其后面的 5 个元素时,标志变量 **flag** 被置为 1 否则标志 **flag** 被置为 0 并且跳出内层 **for** 循环。下面的 **if** 语句的作用是判断 **a[i]** 是否为奇数,若是偶数则将 **flag** 置为 0。**for** 循环结束后的 **if** 语句作用是,通过标志变量 **flag** 来判断 **a[i]** 是否满足题目中的条件,即 **a[i]** 即连续大于后面 5 个数字 又是奇数时 若满足则将 **a[i]** 存入 **b** 数组中。最后对 **b** 数组中的元素进行升序排列,排序完成之后,函数结束。

4. 编写一个函数 **findStr()** 该函数统计一个长度为 2 的子字符串在另一个字符串中出现的次数。例如,假定输入的字符串为 "asdasasdfgasdaszx67asdmklo" 子字符串为 "as" 函数返回值是 6。

函数 **ReadWrite()** 实现从文件 **in4.dat** 中读取两个字符串 并调用函数 **findStr()** 最后把结果输出到文件 **out4.dat** 中。

注意:部分程序已经给出。

请勿改动主函数 **main()** 和其他函数中的任何内容 仅在函数 **findStr()** 的花括号中填入你编写的若干语句。

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int findStr(char *str,char *substr)
{

}

main()
{
    char str[81],substr[3];
    int n;
    clrscr();
    printf("enter the original string:");
    gets(str);
    printf("enter an substring:");
    gets(substr);
    puts(str);
    puts(substr);
    n=findStr(str,substr);
    printf("n=%d\n",n);
    ReadWrite();
}

ReadWrite()
```

```

char str[81], substr[3], ch;
int n, len, i = 0;
FILE * rf, * wf;
rf = fopen("in4.dat", "r");
wf = fopen("out4.dat", "w");
while(i < 5)
{
    fgets(str, 80, rf);
    fgets(substr, 10, rf);
    len = strlen(substr) - 1;
    ch = substr[len];
    if(ch == '\n') substr[len] = 0;
    n = findStr(str, substr);
    fprintf(wf, "%d\n", n);
    i++;
}

fclose(rf);
fclose(wf);

```

【答案】

```
int findStr(char * str, char * substr)
```

```

{
    int n;
    char * p, * r;
    n = 0;
    while(* str)
    {
        p = str;
        r = substr;
        while(* r)
            if(* r == * p)
            {
                r++;
                p++;
            }
        else
            break;
        if(* r == '\0')
            n++;
        str++;
    }

    return n;
}

```

【解析】

考点：(1)在给定的字符串中查找指定字符；

(2)统计查找后满足条件的个数。

本题的解题思路是：对原字符串中的字符进行扫描，若原字符串中含有子字符串，即原字符串的其中一部分与子字符串

完全相同时，统计出个数。判断原字符串是否含有子字符串，可以将原字符串中的元素逐个与子字符串相比较，当两字符串中各个元素都相同且子字符串结束时，则证明原字符串包含一个子字符串。在统计个数的时候需要设置一个记录变量，每当原字符串包含子字符串一次的时候，该记录变量自动加 1。

程序的流程是：首先通过键盘接收两个字符串，其中一个作为原字符串，另一个作为子字符串，然后调用 `findStr()` 函数对两字符串进行比较，处理后的结果由 `ReadWrite()` 函数写回文件 `out4.dat` 中。

在 `findStr()` 函数中，当原字符串不为空，即 `*str` 不为空时进入到外层 `while` 循环中，此时原字符串指针 `str` 和子字符串指针 `substr` 都指向其字符串内的第一个元素，将两字符串的指针分别赋给 `p` 和 `r` 然后再对 `p` 和 `r` 进行比较。这样做的好处是，当 `p` 与 `r` 改变的时候 `str` 和 `substr` 不会受到影响。当子字符串也不为空时即 `*r` 不为空时 进入内嵌的 `while` 循环，该 `while` 循环的功能是，将原字符串与子字符串逐个元素进行比较看是否相等，结束该层 `while` 循环的情况有两种：(1) 比较完毕 即原字符串中包含该子字符串，此时子字符串的指针指向串尾（为 `'\0'`）；(2) 未比较完毕，此时子字符串的指针不指向串尾。下面的 `if` 语句的功能是，通过判断子字符串的指针是否指向串尾，进而来判断内层 `while` 循环结束的原因，若是比较完成，即 `*r == '\0'` 则把记录个数的变量加 1，并且原字符串指针也指向下一个字符，为重新进入外层 `while` 循环作准备。

5. 编写函数 `jsValue()` 它的功能是 求 Fibonacci 数列中大于 `t` 的最小一个数 结果由函数返回。其中 Fibonacci 数列 `F(n)` 的定义为：

$F(0) = 0, F(1) = 1$

$F(n) = F(n-1) + F(n-2)$

最后调用函数 `writeDat()` 读取 50 个数据 `t` 分别得出结果且把结果输出到文件 `out5.dat` 中。

例如 当 `t = 1000` 时 函数值为 :1597。

部分程序已经给出。

请勿改动主函数 `main()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>

int jsValue(int t)
{

}

main()
{
    int n;
    n = 1000;
    printf("n = %d, f = %d\n", n, jsValue(n));
    writeDat();
}

writeDat()
{
    FILE *in, *out;
    int i, n, s;
    in = fopen("in5.dat", "r");
    out = fopen("out5.dat", "w");
    for(i = 0; i < 50; i++)
    {
        fscanf(in, "%d", &n);
        s = jsValue(n);
        printf("%d\n", s);
        fprintf(out, "%d\n", s);
    }
    fclose(in);
}
```

```

        fclose(out);
    }
}

【答案】
int jsValue(int t)
{
    int f1 = 0, f2 = 1, fn;
    fn = f1 + f2;
    while(fn <= t)
    {
        f1 = f2;
        f2 = fn;
        fn = f1 + f2;
    }
    return fn;
}

```

【解析】

考点：(1) 求解 Fibonacci 数列的第 n 项的值；

(2) 查找满足条件的 Fibonacci 数列的第 n 项值。

本题的解题思路是从第一项开始，逐个求出 Fibonacci 数列的每一项的值并且与给定的数据相比较，若找到第一个大于给定的数据则将其返回。

程序的流程是先调用 jsValue(n) 函数处理数据，由 writeDat() 函数将数据写回文件 out5.dat 中。

在 jsValue 函数中，根据题目的条件，Fibonacci 数列中每一项的值均为前两项之和，将 Fibonacci 数列的每一项都初始化，然后通过 while 循环的条件进行判断，当 $fn \leq t$ 时，即第 n 项的值小于 1000，进入 while 循环体，计算 fn 的值，每计算一个 fn 的值就与 t 相比较一次，直到 $fn > t$ 即找到大于 t 的最小的一个数，循环结束，该层 while 循环的功能就是查找比 t 大的 fn 的值，while 循环结束后，利用 return 语句返回 fn 的值，函数结束。

6. 请编写函数 countValue()，它的功能是求 n 以内（不包括 n ）同时能被 3 与 7 整除的所有自然数之和的平方根 s ，并作为函数值返回，最后结果 s 输出到文件 out6.dat 中。

例如若 n 为 1000 时，函数值应为 $s = 153.909064$ 。

部分程序已经给出。

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

```

#include <conio.h>
#include <math.h>
#include <stdio.h>
double countValue(int n)
{
    .
    .
    .
}

main()
{
    clrscr();
    printf("the root during the total of the numbers = %f\n", countValue(1000));
    progReadWrite();
}

progReadWrite()
{
    FILE *fp, *wf;

```

```

int i,n;
float s;
fp = fopen( "in6. dat", "r" );
if( fp == NULL)
{
    printf( "the datafile in6. dat is not exist!" );
    return;
}

wf = fopen( " out6. dat", "w" );
for( i = 0; i < 10; i ++ )
{
    fscanf( fp, "%d\n", &n );
    s = countValue( n );
    fprintf( wf, "%f\n", s );
}

fclose( fp );
fclose( wf );
}

```

【答案】

```

double countValue( int n)
{
    double xy = 0.0;
    int i;
    for( i = 1; i < n; i ++ )
        if( i%3 == 0 && i%7 == 0 )
            xy + = i;
    xy = sqrt( ( double ) xy );
    return xy;
}

```

【解析】

考点 (1) 按条件查找数字 ;

(2) 对查找结果进行再加工。

本题的解题思路是 : 先对 1 到 n 之间的所有数字进行筛选 , 若能被 3 和 7 整除 , 则将其累加到一个临时变量中 , 再通过 sqrt () 函数求出其平方根判断是否能整除 , 可以通过判断该数字模 3 和模 7 结果来实现 若为 0 则能整除。为了保证结果的精确度 在调用 sqrt () 函数的时候 需要将 sqrt () 函数的参数强制转换成双精度类型。此部分考生应该仔细分析。

程序的流程是 : 首先调用 countValue () 函数 并传递给 countValue () 函数一个参数 然后由 progReadWrite () 函数将结果写回文件 out6. dat 中

在 countValue () 函数中 , 首先要求传递一个整型变量作为参数 , 其作用是在该参数范围以内进行查找符合条件的整数 , 然后通过一个 for 循环 , 依次取出比 n 小的各个自然数进行判断 for 循环的结束条件是 i = n , 即所有的数字都查找完成后结束 , if 语句的功能是判断数字是否能被 3 和 7 整除 如果 if 语句的条件成立 , 则将该数累加到临时变量 xy 中 最后通过求平方根函数 sqrt () 求得 xy 的平方根 并存入到 xy 中。在求平方根时 为保证精度 先将 xy 强制转化成双精度实型数据 , 注意强制类型转换的括号不可以省略最后将 xy 返回 , 函数结束。

7. 下列程序的功能是 在三位整数 (100 至 999) 中寻找符合条件的整数并依次从小到大存入数组中 ; 该整数既是完全平方数 又有两个数字相同 例如 144, 676 等。

请编制函数 jsValue () 实现此功能 , 满足该条件的整数的个数是通过所编制的函数返回。

最后调用函数 writeDa () 把结果输出到文件 out7. dat 中。

请勿改动主函数 `main()` 和写函数 `writeDat` 的内容。

```
#include <stdio.h>

jsValue(int bb[])
{

}

main()
{
    int b[20], num;
    num = jsValue(b);
    writeDat(num, b);
    clrscr();
}

writeDat(int num, int b[])
{
    FILE *out;
    int i;
    out = fopen("out7.dat", "w");
    printf("%d\n", num);
    fprintf(out, "%d\n", num);
    for(i = 0; i < num; i++)
    {
        printf("%d\n", b[i]);
        fprintf(out, "%d\n", b[i]);
    }
    fclose(out);
}

【答案】
jsValue(int bb[])
{
    int i, j, k = 0;
    int hun, ten, data;
    for(i = 100; i <= 999; i++)
    {
        j = 10;
        while(j * j <= i)
        {
            if(i == j * j)
            {
                hun = i / 100; data = i - hun * 100;
                ten = data / 10; data = data - ten * 10;
                if(hun == ten || hun == data || ten == data)
                {
                    bb[k] = i; k++;
                }
            }
            j++;
        }
    }
    return k;
}
```

【解析】

考点:(1)在给定的范围内按指定条件进行查找和筛选;

(2)对筛选后的结果进行排序处理。

本题的解题思路是:逐个判断 100~999 中的数字是否为完全平方数,若是完全平方数则将其各个位上的数字分离出来进行比较筛选,若有两个数字相同则将其存入数组 bb 中。判断是否为完全平方数,可以通过检索比其自身小的数的平方值,若有相等的,则是完全平方数,否则不是完全平方数。筛选数据可以通过一个 if 语句来实现。通过此算法处理后,不用再对数组 bb 进行排序,数组 bb 中的元素已经按升序排序。读者可以分析其中的道理。

程序的流程是 首先调用 jsValue() 函数 并将数组 b 作为参数传递给 jsValue() 函数,然后调用 writeDat() 函数将处理的结果写入文件 out7.dat 中

在 jsValue() 函数中 首先通过参数接收数组 b 的首地址,用于以后存放处理好的数据。题目要求查找 100~999 之间的数据 故程序执行 for 循环 (i=1000 时循环结束) 赋 j 的初值为 10,当 $j*j \leq i$,即 i 有可能是一个完全平方数时,while 循环的条件成立 当 $j*j > i$ 时 while 循环结束,即 i 不可能是一个完全平方数。

内嵌 if 语句的功能是进一步判断 i 是否为一个完全平方数,当 if 语句条件成立时,以下四条语句的功能是将数据 i 分解, $i/100$ 得到百位上的数字, $i - \text{hun} * 100$ 得到十位和个位上的数字, $\text{data}/10$ 得到十位上的数字, $\text{data} - \text{ten} * 10$ 得到个位上的数字 接下来 if 语句的功能是对分离后的数字进行判断,如果百位与十位相等或者百位与个位相等,或者十位与个位相等时,即有任意两位数字相等,就把当前数字 i 存入到数组 bb 中 并且 k 加 1,while 循环结束 j 加 1 进入下一次 for 循环。for 循环结束后 将 k 返回。函数结束。

8. 下列程序的功能是:寻找并输出 11 至 999 之间的数 m 它满足 m , m^2 和 m^3 均为回文数。所谓回文数是指各位数字左右对称的整数 例如 121,676,94249 等。满足上述条件的数如 $m = 11, m^2 = 121, m^3 = 1331$ 皆为回文数。请编制函数 int jsValue(long m) 实现此功能 如果是回文数 则函数返回 1 反之则返回 0。最后把结果输出到文件 out8.dat 中。

注意:部分源程序已给出。

请勿改动主函数 main() 的内容。

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

```
int jsValue(long n)
```

```
{
```

```
}
```

```
main()
```

```
{
```

```
    long m;
```

```
    FILE *out;
```

```
    out = fopen("out8.dat", "w");
```

```
    for(m = 11; m < 1000; m++)
```

```
    {
```

```
        if(jsValue(m) && jsValue(m*m) && jsValue(m*m*m))
```

```
        {
```

```
            printf("m = %4ld, m * m = %6ld, m * m * m = %8ld\n", m, m * m, m * m * m);
```

```
            fprintf(out, "m = %4ld, m * m = %6ld, m * m * m = %8ld\n", m, m * m, m * m * m);
```

```
        }
```

```
    }
```

```
    fclose(out);
```

```
    system("pause");
```

```
}
```

【答案】

```
int jsValue(long n)
```

```
{
```

```

int i, strl, half;
char xy[20];
ltoa(n, xy, 10);
strl = strlen(xy);
half = strl/2;
for(i = 0; i < half; i++)
if(xy[i] != xy[strl - i - 1]) break;
if(i >= half) return 1;
else return 0;
}

```

【解析】

考点：判断回文数的算法。

本题的解题思路是在 11 ~ 999 之间进行逐个判断看是否为回文数，若是回文数则判断其平方是否为回文数，若也是回文数则判断其立方是否为回文数，若仍然是回文数则将原数返回。判断是否为回文数的方法是：先将其转化为字符型数据，再将第一个字符与最后一个字符相比较，依此类推，一直执行到中间两个元素比较完成，若该字符串左右对称则是回文数。

程序的流程是：首先打开 out8.dat 文件，然后通过 for 循环来调用 jsValue() 函数，若 m, m * m, m * m * m 都是回文数，即 jsValue() 函数返回值都为 1 时，就将 m, m * m, m * m * m 都写入到 out8.dat 文件中，最后关闭文件 out8.dat。

在 jsValue() 函数中，首先函数接收一个长整型数据作为参数，然后调用 ltoa() 函数。该库函数的格式为：ltoa(参数 1, 参数 2, 参数 3)；参数 1 为一个整型数据，参数 2 为一个字符串的地址；参数 3 为一个整型常量，该函数的功能是将第一个参数转换成字符串，并且存入第二个参数所指的内存单元中。然后通过 strlen() 函数得到数组 xy 的实际长度并将其赋给 strl。下面的 for 循环的功能是：比较第一个字符与最后一个字符，如果不等则跳出循环，若相等则继续比较下一个字符，在比较的过程中每次 i 与 strl 分别向后和向前移动一位，比较一直进行到中间元素被比较过后结束，或者有不相等的元素时结束。

if 语句的功能是确定跳出 for 循环的原因。若为 i >= half 即比较完成，则返回 1，证明该数字是回文数，否则返回 0（即不是回文数）函数结束。

9. 已知在文件 in9.dat 中存在若干个（<200）四位数字的正整数，函数 readDat() 读取这若干个正整数并存入数组 xx 中。请编制函数 CalValue()，其功能要求：(1) 求出这个文件中共有多少个正整数 totNum；(2) 求这些数右移 1 位后，产生的新数是偶数的数的个数 totCnt，以及满足此条件的这些数（右移前的值）的算术平均值 totPjz。最后调用函数 writeDat() 把所求的结果输出到文件 out9.dat 中。

部分程序已经给出。

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

```

#include <stdio.h>
#include <conio.h>
#define MAXNUM 200
int xx[MAXNUM];
int totNum = 0;
int totCnt = 0;
double totPjz = 0.0;
int readDat(void);
void writeDat(void);
void CalValue(void)
{
}

void main()
{
    int i;

```

```

clrscr();
for(i=0;i<MAXNUM;i++);
if(readDat())
{
printf("数据文件 in9.dat 不能打开! \007\n");
return;
}

CalValue();
printf("文件 in9.dat 中共有正整数=%d 个\n",totNum);
printf("符合条件的正整数的个数=%d 个\n",totCnt);
printf("平均值=%.2f\n",totPjz);
writeDat();
}

int readDat(void)
{
FILE *fp;
int i=0;
if((fp=fopen("in9.dat","r"))==NULL)return 1;
while(!feof(fp))
{
fscanf(fp,"%d",&xx[i++]);
}
fclose(fp);
return 0;
}

void writeDat(void)
{
FILE *fp;
fp=fopen("out9.dat","w");
fprintf(fp,"%d\n%d\n%.2f\n",totNum,totCnt,totPjz);
fclose(fp);
}

```

【答案】

```

void CalValue(void)
{
int i,data;
for(i=0;i<MAXNUM;i++)
{
if(!xx[i]) break;
if(xx[i]>0) totNum++;
data=xx[i]>>1;
if(data%2==0)
{totCnt++;totPjz+=xx[i];}
}
totPjz/=totCnt;
}

```

【解析】

考点:(1)统计个数;

(2)移位运算符“>>”的使用;

(3)求算术平均值。

本题的解题思路是:先统计出数组中共有多少个正整数,然后逐个将数组中的元素右移一位,并统计出移位后的数字是偶数的个数,最后求出满足此条件数字的平均值。条件在题目中已经给出,考生应仔细分析这些条件,不应急于开始编程。右移一位可以通过“>>”运算符实现,判断是否为偶数可以通过判断 $xx[i] \% 2$ 的结果是否为零来实现,求平均值需要先将数组中符合条件的数字累加到一个临时变量中,再通过临时变量计算出平均值。

程序的流程是:首先通过 readDat() 函数将数据读入数组 xx 中,然后通过 CalValue() 函数进行处理后由 writeDat() 函数将结果写回文件 out9.dat 中去。

在 CalValue() 函数中,首先执行 for 循环体(i = MAXNUM 循环结束,MAXNUM 为数组中元素的最大个数),内嵌 if 语句的功能是判断 xx 数组中的元素是否为 0。当 xx 数组中的元素为 0 时,则跳出循环,即数组的此位置没有元素。第二个 if 语句的作用是当 xx 数组中的元素为大于 0 时,即数组的该位置有数字时,totNum 加 1,且把 $xx[i]$ 向右移一位后赋给 data,其中“>>”运算符的作用是把 $xx[i]$ 转换成二进制数据后,向右移动一位舍掉一个二进制位,再把该二进制化成十进制。

if 语句的作用是,判断 data 是否是偶数,如果 data 能被 2 整除,即是一个偶数,则总个数 totCnt 加 1,并把 $xx[i]$ 累加到 totPjz 当中去,for 循环结束后,计算 totPjz 的值函数结束。

10 请编制函数 ReadDat() 实现从文件 in10.dat 中读取 1000 个十进制整数到数组 xx 中,请编制函数 Compute() 分别计算出 xx 中偶数的个数 even、奇数的平均值 ave1、偶数的平均值 ave2 以及方差 totfc 的值,最后调用函数 WriteDat() 把结果输出到 out10.dat 文件中。

计算方差的公式如下:

$$\text{totfc} = \frac{1}{N} \sum_{i=0}^{N-1} (xx[i] - \text{ave2})^2$$

设 N 为偶数的个数, $xx[i]$ 为偶数, ave2 为偶数的平均值。

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 2000) 部分程序已经给出。请勿改动主函数 main() 和输出数据函数 WriteDat() 的内容。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 1000

int xx[ MAX ], odd = 0, even = 0;
double ave1 = 0.0, ave2 = 0.0, totfc = 0.0;
void WriteDat( void );
int ReadDat( void )
{
    FILE * fp;
    int i, j;
    if( ( fp = fopen( "in10.dat", "r" ) ) == NULL ) return 1;
    for( i = 0; i < 100; i ++ )
    {
        for( j = 0; j < 10; j ++ )
            fscanf( fp, "%d,", &xx[ i * 10 + j ] );
        fscanf( fp, "\n" );
        if( feof( fp ) ) break;
    }

    fclose( fp );
    return 0;
}
```

```

}

void Compute( void)
{

}

void main( )
{
    int i;
    for( i = 0; i < MAX; i ++ ) xx[ i ] = 0;
    if( ReadDat( ) )
    {
        printf( " 数据文件 in10. dat 不能打开! \007\n" );
        return;
    }
    Compute( ) ;
    printf( " even = %d \n ave1 = %f \n ave2 = %f \n totfc = %f \n" , even , ave1 , ave2 , totfc );
    WriteDat( ) ;
}

void WriteDat( void)
{
    FILE * fp;
    int i;
    fp = fopen( " out10. dat" , " w" );
    fprintf( fp , " %d \n %f \n %f \n %f \n" , even , ave1 , ave2 , totfc );
    fclose( fp );
}

```

【答案】

```

void Compute( void)
{
    int i, yy[ MAX ];
    for( i = 0; i < 1000; i ++ )
        if( xx[ i ] % 2 )
        {
            odd ++ ;
            ave1 + = xx[ i ];
        }
        else
        {
            even ++ ;
            ave2 + = xx[ i ];
            yy[ even - 1 ] = xx[ i ];
        }

    ave1 / = odd;
    ave2 / = even;
    for( i = 0; i < even; i ++ )
        totfc + = ( yy[ i ] - ave2 ) * ( yy[ i ] - ave2 ) / even;
}

```

【解析】

考点:(1)在给定的数据范围内,统计偶数和奇数的平均值;

(2)求方差。

本题的解题思路是:逐个判断数组 `xx` 中的元素的奇偶性,并分别统计出个数且累加到不同的变量中,从而求出平均值。判断奇偶性可以通过判断 `xx[i]%2` 的值来实现 若为 1 则是奇数 若为 0 则为偶数。统计个数可以通过设置一个记录变量,每当条件成立时变量值加 1 来实现。方差的公式已经给出只需转换相应的 C 语言表达式即可。

程序的流程是 首先调用 `ReadDat()` 函数将数据读入 `xx` 数据中 然后通过 `Compute()` 函数对数据进行处理后,由 `WriteDat()` 函数将数据写回文件中去 本题要求写出 `ReadDat()` 函数,考生可以不把此函数作为重点,在三级的上机考试当中,大多数的程序流程是:先从系统中读取一个文件,并存放在指定一个位置,然后要求考生编写程序处理这些数据,由于我们不知道系统中文件的内容和格式,故多数情况下不要求考生编写。对此感兴趣的考生可以分析一下 `ReadDat()` 函数,其中 `feof()` 函数的功能是判断文件指示器是否已达到了文件的结尾。

在 `Compute()` 函数中 首先进入一个 `for` 循环 (`i=1000` 结束)其内嵌 `if` 语句的功能是判断 `a[i]` 的奇偶性 如果 `xx` 数组中的元素不能够被 2 整除,即为奇数时, `odd` 加 1 且把 `xx[i]` 累加到 `ave1` 中 如果数组中的元素能够被 2 整除,即为偶数时 `even` 加 1 且把 `xx[i]` 累加到 `ave2` 中 并把 `xx[i]` 顺序存入数组 `yy` 中 然后通过 `/=` 将得出奇数的平均值 `ave1` 和偶数的平均值 `ave2`。最后一个 `for` 循环的功能是求方差 `totfc`,关于求方差的公式在题目中已经给出,这里只需转换成相应的 C 语言表达式而已 函数结束。

11.某级数的前两项 $A_1=1, A_2=1$ 以后各项具有如下关系:

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

下列程序的功能是 要求依次对于整数 $M=100, 1000$ 和 10000 求出对应的 n 值 使其满足 $S_n < M$ 且 $S_{n+1} \geq M$ 这里 $S_n = A_1 + A_2 + \dots + A_n$ 并依次把 n 值存入数组单元 `b[0]`, `b[1]` 和 `b[2]` 中 请编制 `jsValue()` 函数来实现此功能,最后调用函数 `writeDat()` 把数组 `b[]` 中的值输出到 `out11.dat` 文件中。

请勿改动主函数 `main()` 和写函数 `writeDat()` 的内容。

```
#include <stdio.h>

int b[3];

void jsValue()
{

}

main()
{

    jsValue();
    printf("M=100,n=%d\nM=1000,n=%d\nM=10000,n=%d\n",b[0],b[1],b[2]);
    writeDat();

    writeDat()

}

FILE *fp;
fp = fopen("out11.dat","w");
printf("%d\n%d\n%d\n",b[0],b[1],b[2]);
fprintf(fp,"%d\n%d\n%d\n",b[0],b[1],b[2]);
fclose(fp);

}
```

【答案】

```
void jsValue()
{

    int A1=1,A2=1,n=1,An;
```