目录

[1.1逆序的三位数（5分） 1](#_Toc498617541)

[2.1时间换算（5分） 1](#_Toc498617542)

[2.2信号报告（5分） 2](#_Toc498617543)

[3.1奇偶个数（5分） 3](#_Toc498617544)

[3.2数字特征值（5分） 4](#_Toc498617545)

[4.1素数和（5分） 4](#_Toc498617546)

[4.2念整数（5分） 5](#_Toc498617547)

[5.1高精度小数（10分） 6](#_Toc498617548)

[6.1分解质因数（5分） 7](#_Toc498617549)

[6.2完数（5分） 7](#_Toc498617550)

[7.1多项式加法（5分） 8](#_Toc498617551)

[7.2鞍点（5分） 9](#_Toc498617552)

[8.1单词长度（4分） 10](#_Toc498617553)

[8.2GPS数据处理（6分） 10](#_Toc498617554)

[期末考试 最简分式（10分） 12](#_Toc498617555)

# 1.1逆序的三位数（5分）

**题目内容：**

逆序的三位数：

程序每次读入一个正三位数，然后输出逆序的数字。注意，当输入的数字含有结尾的0时，输出不应带有前导的0。比如输入700，输出应该是7。

提示：用%10可以得到个位数，用/100可以得到百位数...。将这样得到的三个数字合起来：百位\*100+十位\*10+个位，就得到了结果。

***注意：除了题目要求的输出，不能输出任何其他内容，比如输入时的提示，输出时的说明等等都不能。这道题目要求输出逆序的数字，程序就只能输出这个数字，除此之外任何内容都不能输出。***

**输入格式:**

每个测试是一个3位的正整数。

**输出格式：**

输出逆序的数。

**输入样例：**

123

**输出样例：**

321

**时间限制：500ms内存限制：32000kb**

# 2.1时间换算（5分）

**题目内容：**

UTC是世界协调时，BJT是北京时间，UTC时间相当于BJT减去8。现在，你的程序要读入一个整数，表示BJT的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。如1124表示11点24分，而905表示9点5分，36表示0点36分，7表示0点7分。

有效的输入范围是0到2359，即你的程序不可能从测试服务器读到0到2359以外的输入数据。

你的程序要输出这个时间对应的UTC时间，输出的格式和输入的相同，即输出一个整数，表示UTC的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。

提醒：要小心跨日的换算。

**输入格式:**

一个整数，表示BJT的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。

**输出格式：**

一个整数，表示UTC的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。

**输入样例：**

803

**输出样例：**

3

**时间限制：500ms内存限制：32000kb**

# 2.2信号报告（5分）

**题目内容：**

无线电台的RS制信号报告是由三两个部分组成的：

* R(Readability) 信号可辨度即清晰度.
* S(Strength)    信号强度即大小.

其中R位于报告第一位，共分5级，用1—5数字表示.

1. Unreadable
2. Barely readable, occasional words distinguishable
3. Readable with considerable difficulty
4. Readable with practically no difficulty
5. Perfectly readable

报告第二位是S，共分九个级别，用1—9中的一位数字表示

1. Faint signals, barely perceptible
2. Very weak signals
3. Weak signals
4. Fair signals
5. Fairly good signals
6. Good signals
7. Moderately strong signals
8. Strong signals
9. Extremely strong signals

现在，你的程序要读入一个信号报告的数字，然后输出对应的含义。如读到59，则输出：

Extremely strong signals, perfectly readable.

**输入格式:**

一个整数，信号报告。整数的十位部分表示可辨度，个位部分表示强度。输入的整数范围是[11,59]中的个位不为0的数字，这个范围外的数字不可能出现在测试数据中。

**输出格式：**

一句话，表示这个信号报告的意义。按照题目中的文字，先输出表示强度的文字，跟上逗号和空格，然后是表示可辨度的文字，跟上句号。注意可辨度的句子的第一个字母是小写的。注意这里的标点符号都是英文的。

**输入样例：**

33

**输出样例：**

Weak signals, readable with considerable difficulty.

**时间限制：500ms内存限制：32000kb**

# 3.1奇偶个数（5分）

**题目内容：**

你的程序要读入一系列正整数数据，输入-1表示输入结束，-1本身不是输入的数据。程序输出读到的数据中的奇数和偶数的个数。

**输入格式:**

一系列正整数，整数的范围是（0,100000）。如果输入-1则表示输入结束。

**输出格式：**

两个整数，第一个整数表示读入数据中的奇数的个数，第二个整数表示读入数据中的偶数的个数。两个整数之间以空格分隔。

**输入样例：**

9 3 4 2 5 7 －1

**输出样例：**

4 2

**时间限制：500ms内存限制：32000kb**

# 3.2数字特征值（5分）

**题目内容：**

对数字求特征值是常用的编码算法，奇偶特征是一种简单的特征值。对于一个整数，从个位开始对每一位数字编号，个位是1号，十位是2号，以此类推。这个整数在第n位上的数字记作x，如果x和n的奇偶性相同，则记下一个1，否则记下一个0。按照整数的顺序把对应位的表示奇偶性的0和1都记录下来，就形成了一个二进制数字。比如，对于342315，这个二进制数字就是001101。

这里的计算可以用下面的表格来表示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数字 | 3 | 4 | 2 | 3 | 1 | 5 |
| 数位 | 6 | 5 | 4 | 3 | 2 | 1 |
| 数字奇偶 | 奇 | 偶 | 偶 | 奇 | 奇 | 奇 |
| 数位奇偶 | 偶 | 奇 | 偶 | 奇 | 偶 | 奇 |
| 奇偶一致 | 0 | 0 | 1 | 1 | 0 | 1 |
| 二进制位值 | 32 | 16 | 8 | 4 | 2 | 1 |

你的程序要读入一个非负整数，整数的范围是[0,1000000]，然后按照上述算法计算出表示奇偶性的那个二进制数字，输出它对应的十进制值。

***提示：将整数从右向左分解，数位每次加1，而二进制值每次乘2。***

**输入格式:**

一个非负整数，整数的范围是[0,1000000]。

**输出格式：**

一个整数，表示计算结果。

**输入样例：**

342315

**输出样例：**

13

**时间限制：500ms内存限制：32000kb**

# 4.1素数和（5分）

**题目内容：**

我们认为2是第一个素数，3是第二个素数，5是第三个素数，依次类推。

现在，给定两个整数n和m，0<n<=m<=200，你的程序要计算第n个素数到第m个素数之间所有的素数的和，包括第n个素数和第m个素数。

**输入格式:**

两个整数，第一个表示n，第二个表示m。

**输出格式：**

一个整数，表示第n个素数到第m个素数之间所有的素数的和，包括第n个素数和第m个素数。

**输入样例：**

2 4

**输出样例：**

15

**时间限制：500ms内存限制：32000kb**

# 4.2念整数（5分）

**题目内容：**

你的程序要读入一个整数，范围是[-100000,100000]。然后，用汉语拼音将这个整数的每一位输出出来。

如输入1234，则输出：

yi er san si

注意，每个字的拼音之间有一个空格，但是最后的字后面没有空格。当遇到负数时，在输出的开头加上“fu”，如-2341输出为：

fu er san si yi

**输入格式:**

一个整数，范围是[-100000,100000]。

**输出格式：**

表示这个整数的每一位数字的汉语拼音，每一位数字的拼音之间以空格分隔，末尾没有空格。

**输入样例：**

-30

**输出样例：**

fu san ling

**时间限制：500ms内存限制：32000kb**

# 5.1高精度小数（10分）

**题目内容：**

由于计算机内部表达方式的限制，浮点运算都有精度问题，为了得到高精度的计算结果，就需要自己设计实现方法。

(0,1)之间的任何浮点数都可以表达为两个正整数的商，为了表达这样两个数的商，可以将相除的结果以多个整数来表示，每个整数表示结果的一位。即商的第一位用一个整数来表示，第二位用另一个整数来表示，以此类推，就可以输出一个高精度的除法结果了。

如16/19的结果0.8421052631...就可以依次输出8、4、2、1、0、5、2、6、3、1...。

而除法的过程，则可以模仿人工列竖式做除法的方式，先将被除数乘以10，得到一位商以后，将余数乘以10作为下一轮计算的被除数：

    160/19->8余8

    80/19->4余4

    ...

当某次余数为0时，则表明除尽。

现在，请写一个程序，输入一个分数，计算出它的小数形式。无论是否可以除尽，输出最多小数点后200位。

**输入格式:**

形如

    a/b

的两个数，其中10<=a<b<100。也就是说，这个小数一定是小于1的正数。

提示：输入是带着两个数中间的“/”的，所以scanf应采用“%d/%d”这样的输入格式。

**输出格式：**

形如

    0.xxxxxxxxx

的小数，小数点后最多200位。输出结束的时候要带着回车换行。如果a/b是一个有限不循环小数，则输出完所有的有效位就可以了，不需要再输出后面的0来凑满200位。

**输入样例：**

16/19

**输出样例：**

0.84210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684210526315789473684

# 6.1分解质因数（5分）

**题目内容：**

每个非素数（合数）都可以写成几个素数（也可称为质数）相乘的形式，这几个素数就都叫做这个合数的质因数。比如，6可以被分解为2x3，而24可以被分解为2x2x2x3。

现在，你的程序要读入一个[2,100000]范围内的整数，然后输出它的质因数分解式；当读到的就是素数时，输出它本身。

提示：可以用一个函数来判断某数是否是素数。

**输入格式:**

一个整数，范围在[2,100000]内。

**输出格式：**

形如：

n=axbxcxd

或

n=n

所有的符号之间都没有空格，x是小写字母x。abcd这样的数字一定是从小到大排列的。

**输入样例：**

18

**输出样例：**

18=2x3x3

**时间限制：500ms内存限制：32000kb**

# 6.2完数（5分）

**题目内容：**

一个正整数的因子是所有可以整除它的正整数。而一个数如果恰好等于除它本身外的因子之和，这个数就称为完数。例如6=1＋2＋3(6的因子是1,2,3)。

现在，你要写一个程序，读入两个正整数n和m（1<=n<m<1000），输出[n,m]范围内所有的完数。

提示：可以写一个函数来判断某个数是否是完数。

**输入格式:**

两个正整数，以空格分隔。

**输出格式：**

其间所有的完数，以空格分隔，最后一个数字后面没有空格。如果没有，则输出一行文字：

NIL

（输出NIL三个大写字母加回车）。

**输入样例：**

1 10

**输出样例：**

6

**时间限制：500ms内存限制：32000kb**

# 7.1多项式加法（5分）

**题目内容：**

一个多项式可以表达为x的各次幂与系数乘积的和，比如：

2x6+3x5+12x3+6x+20

现在，你的程序要读入两个多项式，然后输出这两个多项式的和，也就是把对应的幂上的系数相加然后输出。

程序要处理的幂最大为100。

**输入格式:**

总共要输入两个多项式，每个多项式的输入格式如下：

每行输入两个数字，第一个表示幂次，第二个表示该幂次的系数，所有的系数都是整数。第一行一定是最高幂，最后一行一定是0次幂。

注意第一行和最后一行之间不一定按照幂次降低顺序排列；如果某个幂次的系数为0，就不出现在输入数据中了；0次幂的系数为0时还是会出现在输入数据中。

**输出格式：**

从最高幂开始依次降到0幂，如：

2x6+3x5+12x3-6x+20

注意其中的x是小写字母x，而且所有的符号之间都没有空格，如果某个幂的系数为0则不需要有那项。

**输入样例：**

6 2

5 3

3 12

1 6

0 20

6 2

5 3

2 12

1 6

0 20

**输出样例：**

4x6+6x5+12x3+12x2+12x+40

**时间限制：500ms内存限制：32000kb**

# 7.2鞍点（5分）

**题目内容：**

给定一个n\*n矩阵A。矩阵A的鞍点是一个位置（i，j），在该位置上的元素是第i行上的最大数，第j列上的最小数。一个矩阵A也可能没有鞍点。

你的任务是找出A的鞍点。

**输入格式:**

输入的第1行是一个正整数n, （1<=n<=100），然后有n行，每一行有n个整数，同一行上两个整数之间有一个或多个空格。

**输出格式：**

对输入的矩阵，如果找到鞍点，就输出其下标。下标为两个数字，第一个数字是行号，第二个数字是列号，均从0开始计数。

如果找不到，就输出

NO

题目所给的数据保证了不会出现多个鞍点。

**输入样例：**

4

1 7 4 1

4 8 3 6

1 6 1 2

0 7 8 9

**输出样例：**

2 1

**时间限制：500ms内存限制：32000kb**

# 8.1单词长度（4分）

**题目内容：**

你的程序要读入一行文本，其中以空格分隔为若干个单词，以‘.’结束。你要输出这行文本中每个单词的长度。这里的单词与语言无关，可以包括各种符号，比如“it's”算一个单词，长度为4。注意，行中可能出现连续的空格。

**输入格式:**

输入在一行中给出一行文本，以‘.’结束，结尾的句号不能计算在最后一个单词的长度内。

**输出格式：**

在一行中输出这行文本对应的单词的长度，每个长度之间以空格隔开，行末没有最后的空格。

**输入样例：**

It's great to see you here.

**输出样例：**

4 5 2 3 3 4

**时间限制：500ms内存限制：32000kb**

# 8.2GPS数据处理（6分）

**题目内容：**

NMEA-0183协议是为了在不同的GPS（全球定位系统）导航设备中建立统一的BTCM（海事无线电技术委员会）标准，由美国国家海洋电子协会（NMEA-The National Marine Electronics Associa-tion）制定的一套通讯协议。GPS接收机根据NMEA-0183协议的标准规范，将位置、速度等信息通过串口传送到PC机、PDA等设备。

NMEA-0183协议是GPS接收机应当遵守的标准协议，也是目前GPS接收机上使用最广泛的协议，大多数常见的GPS接收机、GPS数据处理软件、导航软件都遵守或者至少兼容这个协议。

NMEA-0183协议定义的语句非常多，但是常用的或者说兼容性最广的语句只有$GPGGA、$GPGSA、$GPGSV、$GPRMC、$GPVTG、$GPGLL等。

其中$GPRMC语句的格式如下：

    $GPRMC,024813.640,A,3158.4608,N,11848.3737,E,10.05,324.27,150706,,,A\*50

这里整条语句是一个文本行，行中以逗号“,”隔开各个字段，每个字段的大小（长度）不一，这里的示例只是一种可能，并不能认为字段的大小就如上述例句一样。

    字段0：$GPRMC，语句ID，表明该语句为Recommended Minimum Specific GPS/TRANSIT Data（RMC）推荐最小定位信息

    字段1：UTC时间，hhmmss.sss格式

    字段2：状态，A=定位，V=未定位

    字段3：纬度ddmm.mmmm，度分格式（前导位数不足则补0）

    字段4：纬度N（北纬）或S（南纬）

    字段5：经度dddmm.mmmm，度分格式（前导位数不足则补0）

    字段6：经度E（东经）或W（西经）

    字段7：速度，节，Knots

    字段8：方位角，度

    字段9：UTC日期，DDMMYY格式

    字段10：磁偏角，（000 - 180）度（前导位数不足则补0）

    字段11：磁偏角方向，E=东W=西

    字段16：校验值

这里，“\*”为校验和识别符，其后面的两位数为校验和，代表了“$”和“\*”之间所有字符（不包括这两个字符）的异或值的十六进制值。上面这条例句的校验和是十六进制的50，也就是十进制的80。

提示：^运算符的作用是异或。将$和\*之间所有的字符做^运算(第一个字符和第二个字符异或，结果再和第三个字符异或，依此类推)之后的值对65536取余后的结果，应该和\*后面的两个十六进制数字的值相等，否则的话说明这条语句在传输中发生了错误。注意这个十六进制值中是会出现A-F的大写字母的。

现在，你的程序要读入一系列GPS输出，其中包含$GPRMC，也包含其他语句。在数据的最后，有一行单独的

    END

表示数据的结束。

你的程序要从中找出$GPRMC语句，计算校验和，找出其中校验正确，并且字段2表示已定位的语句，从中计算出时间，换算成北京时间。一次数据中会包含多条$GPRMC语句，以最后一条语句得到的北京时间作为结果输出。

你的程序一定会读到一条有效的$GPRMC语句。

**输入格式:**

多条GPS语句，每条均以回车换行结束。最后一行是END三个大写字母。

**输出格式：**

6位数时间，表达为：

    hh:mm:ss

其中，hh是两位数的小时，不足两位时前面补0；mm是两位数的分钟，不足两位时前面补0；ss是两位数的秒，不足两位时前面补0。

**输入样例：**

$GPRMC,024813.640,A,3158.4608,N,11848.3737,E,10.05,324.27,150706,,,A\*50

END

**输出样例：**

10:48:13

**时间限制：500ms内存限制：32000kb**

# 期末考试 最简分式（10分）

**题目内容：**

分数可以表示为“分子/分母”的形式。编写一个程序，要求用户输入一个分数，然后将其约分为最简分式。最简分式是指分子和分母不具有可以约分的成分了。如6/12可以被约分为1/2。当分子大于分母时，不需要表达为整数又分数的形式，即11/8还是11/8；而当分子分母相等时，仍然表达为1/1的分数形式。

**输入格式:**

输入在一行中给出一个分数，分子和分母中间以斜杠“/”分隔，如： 12/34 表示34分之12。分子和分母都是正整数（不包含0）。

***提示：在scnaf的格式字符串中加入“／”。***

**输出格式：**

在一行中输出这个分数对应的最简分式，格式与输入的相同，即采用“分子/分母”的形式表示分数。如 5/6 表示 6分之5

**输入样例：**

60/120

**输出样例：**

1/2

**时间限制：500ms内存限制：32000kb**