第二次作业

Unix 下数据处理练习：

Cut：

cut 一般格式为：
cut [options] file1 file2
下面介绍其可用选项：
-c list 指定剪切字符数。
-f field 指定剪切域数。
-d 指定与空格和 tab 键不同的域分隔符。
-c 用来指定剪切范围，如下所示：
-c1,5-7 剪切第 1 个字符，然后是第 5 到第 7 个字符。
-c1-50 剪切前 50 个字符。
-f 格式与-c 相同。
-f1,5 剪切第 1 域，第 5 域。
-f1,10-12 剪切第 1 域，第 10 域到第 12 域。

cut –f n inputfile 表示把第 n 个**字符段**剪切出来并显示

cut –f n ,m inputfile 表示把第 n 个和第 m 个字符段剪切并显示

cut –f n-m inputfile 表示把第 n 个到第 m 个字符段剪切并显示

cut –f l,n-m inputfile 表示把第 l 个和第 n 个到第 m 个字符段剪切并显示

cut –c n-m inputfile 表示把第 n 个到第 m 个**字符**剪切出来并显示

同样的练习**-c** 的用法。

Paste：

用 paste 可以将一些数据粘贴起来形成相关文件。
粘贴两个不同来源的数据时，首先需将其分类，并确保两个文件行数相同。paste 将按行将不同文件行信息放在一行。缺省情况下， paste 连接时，用空格或 tab 键分隔新行中不同文

本，除非指定-d 选项，它将成为域分隔符。
paste 格式为:
paste -d -s -file1 file2
选项含义如下：
-d 指定不同于空格或 tab 键的域分隔符。例如用@分隔域，使用- d @。
-s 将每个文件合并成行而不是按行粘贴。
- 使用标准输入。例如，ls -l |paste 意即只在一列上显示输出。

paste inputfile1 inputfile2

paste –d: inputflie1 inputfile2

paste –s inputflie1 inputfile2

ls -l |paste

Sed：sed ‘ 操作的内容’文件名

，使用模式和行号进行查询：查找文件中的相应行号的文本 sed -n ‘行号 p’文件名

查找文件中的包含相应文本行的文本 sed –n ‘/文本/’ 文件名

，插入文本：在选定的行的最前面插入目标文本 sed ‘/文本/i\目标文本’文件名

，附加文本：在选定的行最后面附加目标文本 sed ‘/文本/a\目标文本’ 文件名

，修改文本：对选定的行进行修改成目标文本 sed ‘/文本/c\目标文本’ 文件名

，删除文本：删除所选的行 sed ‘/文本/d’ 文件名

，替换文本：把所选的文本改成目标文本 sed ‘s/文本/目标文本/’ 文件名

附：其中/文本/是用来确定所要修改内容所在的行，所以当然其可以直接用行号来代替。

替换文本一项中/文本/确是直接所要修改的内容而非修改行，如果要全文件对某一文本进行修改则用 ：sed ‘s/文本/目标文本/g’ 文件名

Awk：使用 awk 的第一个理由是基于文本的样式扫描和处理是我们经常做的工作，awk 所做的工作有些象数据库，但与数据库不同的是，它处理的是文本文件，这些文件没有专门的存储格式，普通的人们就能编辑、阅读、理解和处理它们。而数据库文件往往具有特殊的存储格式，这使得它们必须用数据库处理程序来处理它们。

awk 提供两种变量，一种是 awk 内置的变量，这前面我们已经讲过，需要着重指出的是，与后面提到的其它变量不同的是，在 awk 程序中引用内置变量不需要使用标志符"$"（回忆一下前面讲过的 NR 的使用）。awk 提供的另一种变量是自定义变量。awk 允许用户在 awk 程序语句中定义并调用自己的变量。当然这种变量不能与内置变量及其它 awk 保留字相同，在 awk 中引用自定义变量必须在它前面加上标志符"$"。与 C 语言不同的是，awk 中不需要对变量进行初始化，awk 根据其在 awk 中第一次出现的形式和上下文确定其具体的数据类型。当变量类型不确定时，awk 默认其为字符串类型。这里有一个技巧：如果你要让你的 awk 程序知道你所使用的变量的明确类型，你应当在在程序中给它赋初值

Awk 中内置变量及其含义：

FILENAME 当前输入文件的名字

FS 输入字段分隔符 空格

NF 当前记录中的字段个数

NR 已经读出的记录数

OFMT 数字的输出格式 %.6g

OFS 输出字段分隔符 空格

ORS 输出的记录分隔符 新行

RS 输入的记录他隔符 新行

与其它 UNIX 命令一样，awk 拥有自己的语法：

awk [-F re] [parameter...] ['prog'] [-f progfile][in_file...]

参数说明：

-F re:允许 awk 更改其字段分隔符。

parameter: 该参数帮助为不同的变量赋值。

'prog': awk 的程序语句段。这个语句段必须用单拓号：'和'括起，以防被 shell 解释。这个程序语句段的标准形式为：

'pattern {action}'

() ,{}, “”, ‘’ 这四个符号的用处要弄明白。

其中 pattern 参数可以是 egrep 正则表达式中的任何一个，它可以使用语法/re/再加上一些样式匹配技巧构成。与 sed 类似，你也可以使用","分开两样式以选择某个范围。关于匹配的细节，你可以参考附录，如果仍不懂的话，找本 UNIX 书学学 grep 和 sed。action 参数总是被大括号包围，它由一系统 awk 语句组成，各语句之间用";"分隔。awk 解释它们，并在 pattern 给定的样式匹配的记录上执行其操作。与 shell 类似，你也可以使用“#”作为注释符，它使“#”到行尾的内容成为注释，在解释执行时，它们将被忽略。你可以省略 pattern 和 action 之一，但不能两者同时省略，当省略 pattern 时没有样式匹配，表示对所有行（记录）均执行操作，省略 action 时执行缺省的操作——在标准输出上显示。

-f progfile:允许 awk 调用并执行 progfile 指定有程序文件。progfile 是一个文本文件，他必须符合 awk 的语法。

in_file:awk 的输入文件，awk 允许对多个输入文件进行处理。值得注意的是 awk 不修改输入文件。如果未指定输入文件，awk 将接受标准输入，并将结果显示在标准输出上。awk 支持输入输出重定向。

我们一下面的文本文件来练习上面的 cut， paste，sed 三个命令，已达到认识其功能的目的，把文件一改写成文件二。

第一列表示每一行都是一个原子，

第二列原子的序号

第三列原子在特定的力场数据库下的记号

第四列原子所在的氨基酸的简称

第五列原子所在氨基酸的序号

第六至八时原子的 x，y，z 坐标

第九至十列是其它信息

第十一列表示原子名称

文件一：

ATOM	1	N	PHE	2	1.378	-1.795	-1.043	1.00	0.00	N
ATOM	2	CA	PHE	2	1.893	-0.429	-1.043	1.00	0.00	C
ATOM	3	C	PHE	2	3.403	-0.419	-1.043	1.00	0.00	C
ATOM	4	O	PHE	2	4.076	-1.172	-0.286	1.00	0.00	O
ATOM	5	CB	PHE	2	1.385	0.334	0.218	1.00	0.00	C

```
ATOM      6  CG  PHE     2      -0.139   0.429   0.393  1.00  0.00           C
ATOM      7  CD1 PHE     2      -0.821  -0.586   1.074  1.00  0.00           C
ATOM      8  CD2 PHE     2      -0.860   1.483  -0.173  1.00  0.00           C
ATOM      9  CE1 PHE     2      -2.208  -0.556   1.171  1.00  0.00           C
ATOM     10  CE2 PHE     2      -2.248   1.518  -0.066  1.00  0.00           C
ATOM     11  CZ  PHE     2      -2.921   0.497   0.603  1.00  0.00           C
ATOM     12  HA  PHE     2       1.547   0.085  -1.960  1.00  0.00           H
ATOM     13  HB2 PHE     2       1.801   1.361   0.219  1.00  0.00           H
ATOM     14  HB1 PHE     2       1.822  -0.122   1.130  1.00  0.00           H
ATOM     15  HD1 PHE     2      -0.276  -1.423   1.490  1.00  0.00           H
ATOM     16  2HD PHE     2      -0.346   2.267  -0.712  1.00  0.00           H
ATOM     17  HE1 PHE     2      -2.727  -1.360   1.672  1.00  0.00           H
ATOM     18  HE2 PHE     2      -2.802   2.331  -0.512  1.00  0.00           H
ATOM     19  HZ  PHE     2      -3.998   0.515   0.675  1.00  0.00           H
ATOM     20  H   PHE     2       1.259  -2.110  -0.091  1.00  0.00           H
ATOM     21  H   PHE     2       2.030  -2.401  -1.521  1.00  0.00           H
ATOM     22  N   PHE     3       4.111   0.495  -1.952  1.00  0.00           N
ATOM     23  CA  PHE     3       5.536   0.277  -1.725  1.00  0.00           C
ATOM     24  C   PHE     3       6.291   1.585  -1.720  1.00  0.00           C
ATOM     25  O   PHE     3       5.709   2.692  -1.895  1.00  0.00           O
ATOM     26  CB  PHE     3       6.122  -0.638  -2.843  1.00  0.00           C
ATOM     27  CG  PHE     3       5.476  -2.023  -3.002  1.00  0.00           C
ATOM     28  CD1 PHE     3       4.368  -2.176  -3.844  1.00  0.00           C
ATOM     29  CD2 PHE     3       5.941  -3.119  -2.271  1.00  0.00           C
ATOM     30  CE1 PHE     3       3.722  -3.404  -3.937  1.00  0.00           C
ATOM     31  CE2 PHE     3       5.301  -4.351  -2.373  1.00  0.00           C
ATOM     32  CZ  PHE     3       4.190  -4.492  -3.204  1.00  0.00           C
ATOM     33  HA  PHE     3       5.673  -0.201  -0.736  1.00  0.00           H
ATOM     34  HB2 PHE     3       7.208  -0.777  -2.672  1.00  0.00           H
ATOM     35  HB1 PHE     3       6.080  -0.109  -3.817  1.00  0.00           H
ATOM     36  HD1 PHE     3       3.980  -1.328  -4.392  1.00  0.00           H
ATOM     37  HD2 PHE     3       6.789  -3.011  -1.608  1.00  0.00           H
ATOM     38  HE1 PHE     3       2.850  -3.504  -4.566  1.00  0.00           H
ATOM     39  HE2 PHE     3       5.659  -5.193  -1.800  1.00  0.00           H
ATOM     40  HZ  PHE     3       3.684  -5.443  -3.274  1.00  0.00           H
ATOM     41  H   PHE     3       3.694   1.144  -2.604  1.00  0.00           H
END
```

文件二：

```
PHE1      1    N    2.378    0.205    1.957
```

The fisrt residue
PHE1 2 C 2.893 1.571 1.957
The fisrt residue
PHE1 3 C 4.403 1.581 1.957
The fisrt residue
PHE1 4 O 5.076 0.828 2.714
The fisrt residue
PHE1 5 CB 2.385 2.334 3.218
The fisrt residue
PHE1 6 CG 0.861 2.429 3.393
The fisrt residue
PHE1 7 CD 0.179 1.414 4.074
The fisrt residue
PHE1 8 CD 0.14 3.483 2.827
The fisrt residue
PHE1 9 CE -1.208 1.444 4.171
The fisrt residue
PHE1 10 CE -1.248 3.518 2.934
The fisrt residue
PHE1 11 CZ -1.921 2.497 3.603
The fisrt residue
PHE1 12 HA 2.547 2.085 1.04
The fisrt residue
PHE1 13 HB2 2.801 3.361 3.219
The fisrt residue
PHE1 14 HB1 2.822 1.878 4.13
The fisrt residue
PHE1 15 HD1 0.724 0.577 4.49
The fisrt residue
PHE1 16 HD 0.654 4.267 2.288
The fisrt residue
PHE1 17 HE1 -1.727 0.64 4.672
The fisrt residue
PHE1 18 HE2 -1.802 4.331 2.488
The fisrt residue
PHE1 19 HZ -2.998 2.515 3.675
The fisrt residue
PHE1 20 H 2.259 -0.11 2.909
The fisrt residue
PHE1 21 H 3.03 -0.401 1.479
The fisrt residue
PHE2 22 N 5.111 2.495 1.048
The second residue
PHE2 23 CA 6.536 2.277 1.275

The second residue
PHE2 24 C 7.291 3.585 1.28
The second residue
PHE2 25 O 6.709 4.692 1.105
The second residue
PHE2 26 CB 7.122 1.362 0.157
The second residue
PHE2 27 CG 6.476 -0.023 -0.002
The second residue
PHE2 28 CD 5.368 -0.176 -0.844
The second residue
PHE2 29 CD 6.941 -1.119 0.729
The second residue
PHE2 30 CE 4.722 -1.404 -0.937
The second residue
PHE2 31 CE 6.301 -2.351 0.627
The second residue
PHE2 32 CZ 5.19 -2.492 -0.204
The second residue
PHE2 33 HA 6.673 1.799 2.264
The second residue
PHE2 34 HB2 8.208 1.223 0.328
The second residue
PHE2 35 HB1 7.08 1.891 -0.817
The second residue
PHE2 36 HD1 4.98 0.672 -1.392
The second residue
PHE2 37 HD2 7.789 -1.011 1.392
The second residue
PHE2 38 HE1 3.85 -1.504 -1.566
The second residue
PHE2 39 HE2 6.659 -3.193 1.2
The second residue
PHE2 40 HZ 4.684 -3.443 -0.274
The second residue
PHE2 41 H 4.694 3.144 0.396
The second residue

Awk 的练习，利用上面给的文件进行练习：

awk -F % 'NR==7,NR==15 {printf $1 $3 $7}' 显示文本文件 myfile 中第七行到第十五行中以字符%分隔的第一字段，第三字段和第七字段：

awk '/101/' file 显示文件 file 中包含 101 的匹配行。

awk '{print NR,NF,$1,$NF,}' file 显示文件 file 的当前记录号、域数和每一行的第一个和最后一个域

awk -F "|" '{print $1}' file 按照新的分隔符“|”进行操作

awk 'length($0)>80 {print NR}' myfile 显示文本 myfile 中所有超过 80 个字符的行号，在这里，用$0 表示整个记录（行），同时，内置变量 NR 不使用标志符'$

awk 'BEGIN { max=100 ;print "max=" max} BEGIN 表示在处理任意行之前进行的操作

awk ‘END {print $1}’ END 之后列出的操作将在扫描完全部的输入之后执行

awk ‘{print length($1)}’ inputfile 调用内置函数 length()

awk ‘BEGIN {total=0} {print $2 ;total=total+$2} END{print total} inputfile 编一小程序对文件的第二列进行统计求和