

1. 生成 1000 行的文件，每一行可以包含字母、数字。长度不限制。
- \* 按照字母顺序排序这个文件，输出所有的 unique 行。
  - \* 以覆盖写的方式将上述的结果重定向到一个文件中。

实现：

创建一个文档 String.sh，内容如下：

```
1 for i in {1..1000}; do
2 echo $RANDOM | md5sum | cut -c 1-15 >> Draft1.txt
3 done
4 sort -u Draft1.txt > StringResult.txt
```

在 Terminal 处，运行如下指令：

```
s10215501434@JupyterHub:~$ ./String.sh
```

即可生成一个 Draft1.txt 文件，其中内容为 1000 个由随机数进行 MD5 加密后得到的随机字符串的前 15 位。Draft1.txt 文件生成完毕后，将会对其中的文字进行排序，并且将去重的排序结果以覆盖的方式写入 StringResult.txt。

这样的操作实现了“按照字母顺序排序这个文件，输出所有的 unique 行”与“以覆盖写的方式将上述的结果重定向到一个文件中”的两个目的。

---

- \* 按照数字顺序排序这个文件，输出所有的 unique 行。
- \* 以追加写的方式将上述的结果重定向到一个文件中

实现：

创建一个 Number.sh，内容如下：

```
1 for i in {1..1000}; do
2 echo $RANDOM >> Draft2.txt
3 done
4 sort -n -u Draft2.txt >> NumberResult.txt
```

在 Terminal 处，运行如下指令：

```
s10215501434@JupyterHub:~$ ./Number.sh
```

即可生成一个 Draft2.txt 文件，其中内容为 1000 个随机数。Draft2.txt 文件生成完毕后，将会对其中的数字按照数字大小进行排序，并且将去重的排序结果以追加的方式写入 NumberResult.txt。

这样的操作实现了“按照数字顺序排序这个文件，输出所有的 unique 行”与“以追加写的方式将上述的结果重定向到一个文件中”的两个目的。

2. 随便生成一个文件，我们想要在这个文件中做以下几件事情：

- \* 找一个内容确定的字符串，比如 “computer”
- \* 统计其中包含多少 computer

实现：

假设以歌曲《Your Reality》的歌词为内容的文件 Your\_Reality.txt 为随便生成的文件，现统计其中出现的单词 “you” 及其出现的次数。

通过以下指令，我们可以直接看到在文件中的哪些地方含有单词 “you”（不区分大小写）：

```
s10215501434@JupyterHub:~$ grep -i you Your_Reality.txt
```

结果示例：

```
Every day, I imagine a future where I can be with you  
In my hand is a pen that will write a poem of me and you  
Just move your hand - write the way into his heart!  
When you're here, everything that we do is fun for them anyway  
Is it love if I take you, or is it love if I set you free?  
If I can't hear the sound of your heartbeat  
What do you call love in your reality?  
And in your reality, if I don't know how to love you  
I'll leave you be
```

通过以下指令，我们可以直接看到在文件中出现了多少次单词 “you”（不区分大小写）：

```
s10215501434@JupyterHub:~$ grep -i -o you Your_Reality.txt | wc -l
```

结果为 12.