

profilng1/concord.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # concord.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import re
10:
11: #-----
12:
13: def process_line(line, concordance):
14:     line = line.lower()
15:     re_letters = re.compile(r'[a-z]+')
16:     words = re_letters.findall(line)
17:     for word in words:
18:         if word in concordance:
19:             concordance[word] += 1
20:         else:
21:             concordance[word] = 1
22:
23: #-----
24:
25: def main():
26:
27:     concordance = {}
28:     for line in sys.stdin:
29:         process_line(line, concordance)
30:     #for word in concordance:
31:     #    print('%s: %d' % (word, concordance[word]))
32:     for word, count in concordance.items():
33:         print('%s: %d' % (word, count))
34:
35: #-----
36:
37: if __name__ == '__main__':
38:     main()

```

profilng1/writeprofile.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # writeprofile.py
5: # Author: Bob Dondero
6: #-----
7:
8: import pstats
9: import sys
10: p = pstats.Stats(sys.argv[1])
11: p.sort_stats('tottime')
12: p.print_stats()

```

profiling1/buildandrun (Page 1 of 1)

```
1: #!/usr/bin/env bash
2:
3: #-----
4: # buildandrun
5: # Author: Bob Dondero
6: #-----
7:
8: set -o verbose
9:
10: # Create concord.profile
11: python -m cProfile -o concord.profile concord.py < Bible.txt
12:
13: # Generate the report
14: python writeprofile.py concord.profile > report.txt
15:
16: # To view the report examine the contents of report.txt
```

profiling1/buildandrun.bat (Page 1 of 1)

```
1: @ECHO OFF
2: REM -----
3: REM buildandrun
4: REM Author: Bob Dondero
5: REM -----
6:
7: REM Create concord.profile
8: python -m cProfile -o concord.profile concord.py < Bible.txt
9:
10: REM Generate the report
11: python writeprofile.py concord.profile > report.txt
12:
13: REM To view the report examine the contents of report.txt
```

profiling2/concord.py (Page 1 of 1)

```
1: #!/usr/bin/env python
2:
3: #-----
4: # concord.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import re
10:
11: #-----
12:
13: def process_line(line, concordance, re_letters):
14:     line = line.lower()
15:     words = re_letters.findall(line)
16:     for word in words:
17:         if word in concordance:
18:             concordance[word] += 1
19:         else:
20:             concordance[word] = 1
21:
22: #-----
23:
24: def main():
25:
26:     concordance = {}
27:     re_letters = re.compile(r'[a-z]+')
28:     for line in sys.stdin:
29:         process_line(line, concordance, re_letters)
30:     #for word in concordance:
31:     #    print('%s: %d' % (word, concordance[word]))
32:     for word, count in concordance.items():
33:         print('%s: %d' % (word, count))
34:
35: #-----
36:
37: if __name__ == '__main__':
38:     main()
```