

SciComp with Py

Pipes & Pipelines

Vladimir Kulyukin
Department of Computer Science
Utah State University



Pipes

- A Pipe is a stream b/w two processes
- A pipe can be uni-directional or bi-directional
- Pipes originated on Unix and are now available on many other operating systems
- Piping paradigm: write many small useful programs and then use pipes to channel the output of one program as input into another program



Pipelines

- If there are two programs P_1 and P_2 , you can use piping to channel the output of P_1 into P_2 as follows:

> $P_1 | P_2$

- Pipelines can be arbitrarily long:

> $P_1 | P_2 | \dots | P_{n-1} | P_n$



Problem

Let's write a few Py programs for filtering odd/even numbers from standard input (STDIN), thresholding numbers from STDIN, finding min/max of numbers from STDIN. We'll construct different pipelines from these programs.



Filtering Odds & Evens from STDIN

```
from __future__ import print_function
import sys
for n in [int(x) for x in sys.stdin.readlines() if int(x) % 2 != 0]:
    print(n)
```

Py source in filter_stdin_odds.py

```
from __future__ import print_function
import sys
for n in [int(x) for x in sys.stdin.readlines() if int(x) % 2 == 0]:
    print(n)
```

Py source in filter_stdin_evens.py



Testing FILTER_STDIN_EVENS/ODDS

```
$ more numbers2.txt | python filter_stdin_evens.py
```

```
4  
10  
100  
12  
78  
490  
56  
8  
90
```

```
$ more numbers2.txt | python filter_stdin_odds.py
```

```
1  
3  
9  
11  
25  
5  
7  
17  
31  
45
```



Scripts for Thresholding STDIN on \leq

```
from __future__ import print_function
import sys
thresh = int(sys.argv[1])
for n in [int(x) for x in sys.stdin.readlines() if int(x) <= thresh]:
    print(n)
```

Py source in `lte_stdin_thresh.py`



Testing LTE_STDIN_TRESH.PY

```
$ cat numbers.txt | python lte_stdin_thresh.py 5
```

```
1  
2  
3  
4  
5
```

```
$ cat numbers2.txt | python lte_stdin_thresh.py 20
```

```
1  
3  
4  
10  
12  
9  
11  
5  
8  
7  
17
```



Scripts for Thresholding STDIN on \geq

```
from __future__ import print_function
import sys
thresh = int(sys.argv[1])
for n in [int(x) for x in sys.stdin.readlines() if int(x)  $\geq$  thresh]:
    print(n)
```

Py source in `gte_stdin_thresh.py`



Testing GTE_STDIN_TRESH.PY

```
$ cat numbers.txt | python gte_stdin_thresh.py 5
```

5

6

7

8

9

10

```
$ cat numbers2.txt | python gte_stdin_thresh.py 20
```

100

25

78

490

56

31

90

47



Finding MAX/MIN in STDIN

```
from __future__ import print_function
import sys
print(max([int(x) for x in sys.stdin.readlines()]))
```

Py source in stdin_max.py

```
from __future__ import print_function
import sys
print(min([int(x) for x in sys.stdin.readlines()]))
```

Py source in stdin_min.py



Problem

Construct a pipeline that find the smallest even number in a file of numbers.



Solution

```
$ more numbers2.txt | python filter_stdin_evens.py | python stdin_min.py  
4
```

OR

```
$ more numbers2.txt | ./filter_stdin_evens.py | ./stdin_min.py  
4
```



Problem

What does the following pipeline compute?

```
$ more numbers2.txt | ./filter_stdin_odds.py | ./stdin_max.py
```

