UNIX PIPES - EXT 0.1

Process Substitution

The cat, sort, and other commands can work with multiple files, for example:

```
$ cat 03.txt 03.txt
```

is going to print the contents of the file 03.txt 2 times.

The shell also allows you to use commands instead of files.

For example:

```
$ cat <(cat 03.txt | head -2) \
      <(cat 03.txt | tail -3)
```

The shell will run cat | head and stream its output to /dev/fd/11, and cat | tail and stream to /dev/fd/12, 11 and 12 are made on demand by the shell.

After it prepares the streams, it starts cat with those streams as arguments:

```
$ cat /dev/fd/11 /dev/fd/12
```

So the result of the command will print the first and the last lines of 03.txt

----------> 02 <----------

This game requires The UNIX Pipes Game version 0, you can get it at:

https://punkx.org/unix-pipe-game/

It contains sort, cat and other commands needed in order to solve the tasks in this game expansion.

The rules are the same:

> 1. The youngest players picks a task from the tasks card. You can not pick the same task twice.

> 2. Shuffle the cards from both decks.

> 3. Put the cards face down on the table.

> 4. Going clockwise each player picks the top card from the deck and try to complete the task

> 5. The first player who completes the task gets a point.

> 6. IF there are no more tasks, GOTO 8

> 7. GOTO 1.

> 8. GAME OVER. INSERT COIN. GOTO 8

-------------------> 03 <--------------------

```
Rank,Title,Year,Rating,Votes
2,Prometheus,2012,7,485820
3,Split,2016,7.3,157606
4,Sing,2016,7.2,60545
5,Suicide Squad,2016,6.2,393727
6,The Great Wall,2016,6.1,56036
7,La La Land,2016,8.3,258682
8,Mindhorn,2016,6.4,2490
9,The Lost City of Z,2016,7.1,7188
10,Passengers,2016,7,192177
12,Hidden Figures,2016,7.8,93103
13,Rogue One,2016,7.9,323118
14,Moana,2016,7.7,118151
15,Colossal,2016,6.4,8612
17,Hacksaw Ridge,2016,8.2,211760
18,Jason Bourne,2016,6.7,150823
19,Lion,2016,8.1,102061
20,Arrival,2016,8,340798
21,Gold,2016,6.7,19053
23,Hounds of Love,2016,6.7,1115
24,Trolls,2016,6.5,38552
26,Paris pieds nus,2016,6.8,222
28,Dead Awake,2016,4.7,523
29,Bad Moms,2016,6.2,66540
30,Assassin's Creed,2016,5.9,112813
31,Why Him?,2016,6.3,48123
32,Nocturnal Animals,2016,7.5,126030
33,X-Men: Apocalypse,2016,7.1,275510
34,Deadpool,2016,8,627797
37,Interstellar,2014,8.6,1047747
38,Doctor Strange,2016,7.6,293732
```

----------> 04 <----------

TASKS

* swap the rating and title column
* print the last row's votes
* remove the Rank and Votes column
* print only the title column uppercased
* print the first and the last movie
* print the votes difference between the top and bottom voted movie
* sum the votes
* sum the votes of movies from 2016
* show the best ranked title, only title, no other columns
* replace commas with tabs, remove space
* print the titles but as a giant string: Sing Interstellar ...
* compute the average rating sum(rating)/(amount of movies)

/usr/bin/paste

`paste` is one of the most underused commands, and it has incredible power, this card is not enough to give it justice, so be sure to read `man paste`.

Imagine you have 3 files:

```
03.txt    04.txt    05.txt
foo       qux       quux
bar       fred      corge
baz       thud      waldo
```

and you want to make a single output merging each line of each file with a comma:

```
$ paste -d, 03.txt 04.txt 05.txt
foo,qux,quux
bar,fred,corge
baz,thud,waldo
```

the -s options processes one file at a time and apply the delimiter per line:

```
$ paste -sd, 03.txt 04.txt 05.txt
foo,bar,baz
qux,fred,thud
quux,corge,waldo
```

/usr/bin/cut

`cut` is incredible tool, it does what it says, it cuts the stream. Make sure to see its man page with `man cut`.

In this example we will use cut on a file example.txt containing:

```
hello world
this is the best
unix pipe game
ever
```

```
$ cat example.txt | cut -f 1 -d ' '
or
$ cut -f 1 -d ' ' example.txt
```

will print the following:

```
hello
this
unix
pipe
ever
```

```
$ cat example.txt | cut -f 2 -d ' '
world
is
pipe
```

/usr/bin/tr

`tr` is a tool to help you transform text, you can change and delete anything use `man tr` to see its potential!

imagine file example.txt containing:

```
99 Bottles on the wall!
98 Bottles.
97 Bottles to rule them all.
96 Bottles to bind them.
```

```
$ cat example.txt | tr '[:digit:]' 'a'
aa Bottles on the wall!
aa Bottles.
aa Bottles to rule them all.
aa Bottles to bind them.
```

```
$ cat example.txt | tr '[a-z]' '[A-Z]'
99 BOTTLES ON THE WALL!
98 BOTTLES.
97 BOTTLES TO RULE THEM ALL.
96 BOTTLES TO BIND THEM.
```

```
$ cat example.txt | tr -d '\n'
99 Bottles on the wall!98 Bottles.97 B..
```

```
$ cat example.txt | tr -d '[bB]'
99 ottles on the wall!
...
96 ottles to ind them.
```

/usr/bin/bc

`bc` is actually a calculator language
but I call it 'bizarre calculator'.

Imagine you have a file 03.txt having:

```
1
3
6
```

If I convert the new lines to '+' for
example using `paste`:

```
$ cat 03.txt | paste -sd+
1+3+6
```

So now we have 1+3+6, if we pipe
that into bc, it will do what you think

```
$ cat 03.txt | paste -sd+ | bc
10
```

As expected the result of 1+3+6 is 10.

To compute 1/3/6 we use delimiter /

```
$ cat 03.txt | paste -sd/ | bc -l
0.05555555555555555555
```

bc -l uses scale=20, as bc is arbitrary
precision you can specify how precise
you want to be, by default its scale=0.

OTHER COMMANDS

The UNIX-like command line is absolutely amazing. A lot of those programs are 50 years old. The concept of pipes and text communication between programs is so powerful that it is still one of the most productive ways to use a computer today.

Check out the following to get started:

```
$ man cut
$ man sed
$ man awk
$ man tr
$ man grep
$ man head
$ man tail
$ man paste
$ man sort
$ man uniq
$ man cat
$ man tac
$ man bc
$ man find
$ man ls
$ man shuf
$ man split
$ man tar
$ man gz
$ man nc
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```

```
| bc -l
```

```
| bc
```

```
----------------------------------------
bc: calculator language
bc -l: default to scale=20
```

```
| paste -dDELIMITERS
```

```
| paste -sdDELIMITERS
```

```
----------------------------------------
paste: merge lines of files
paste -dxz use x and then z as delimiter
paste -s paste one file at a time
```

```
| cut -fFIELDS -dDELIMITER
```

```
----------------------------------------
cut: cut the line
```

```
| tr -d 'SET1'
```

```
| tr 'SET1' 'SET2'
```

```
----------------------------------------
tr: transform or delete characters
tr SET1 SET2: transform SET1 to SET2
tr -d SET1: delete characters in SET1
```