



Introduction To Bioinformatics

Exercise 2

Name: Aris Podotas

University: National and Kapodistrian University of Athens

Program: Data Science and Information Technologies Master's Program

Specialty: Bioinformatics and Biomedical data

Lesson: Bioinformatics

File type: pdf

Table of Contents

Analysis.....	2
A priori setup.....	2
First requirement.....	3
Second requirement	3
Third requirement	4
Fourth requirement	4
Fifth requirement.....	5
Sixth requirement.....	6
Seventh requirement	7
Eighth requirement	7
Nineth requirement	8
Eleventh requirement.....	9
Twelfth requirement	11
Thirteenth requirement	12
Fourteenth requirement	12
Fifteenth requirement.....	13
Explanation of solution:	14

Analysis

A priori setup

Linux availability: Windows Subsystem for Linux (WSL) Ubuntu.

Note: Linux comes with a local installation of vim, a modile text editor that Linus Torvalds must love because he embeds it within his programs (Linux, Git). I share this admiration for vim and its cousin Neovim and thus images and commands will be run within the local vim installation that comes pre embedded within Linux and WSL.

Note: Typing “sudo” all the time is a little annoying and we are on WSL, meaning that the worst-case scenario is having to re download WSL distributions. Thus `''sudo -s''` will be run so that root user privilege is granted.

Note: Sequential requirements will be executed from the exact same point as the previous requirements end point.

Before any of the analysis, this is what the file structure looks like within WSL.

```

aris@LAPTOP-Q684C5C5: ~
=====
Netrw Directory Listing                                (netrw v171)
/home
Sorted by      name
Sort sequence: [\\/]$, \<core\%(\\.ld\+\\)=\>, \.h$, \.c$, \.cpp$, \-|=|*$, *, \.c$, \.obj$, \.info$, \.swp$, \.bak$, \-$.
Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:special
=====
../
./
aris/

```

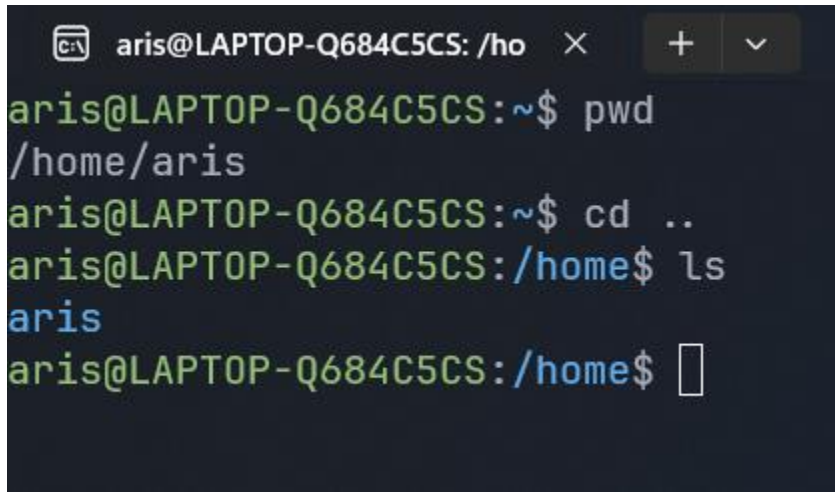
Figure 1: File and folder setup in the `./home /` directory of WSL. The view is from within `vim`

Note: For all the requirements, the process of fulfilling the requirement will be shown. This means that manual pages searched, trial and error and any other such familiarization of the commands will be included in the outputs.



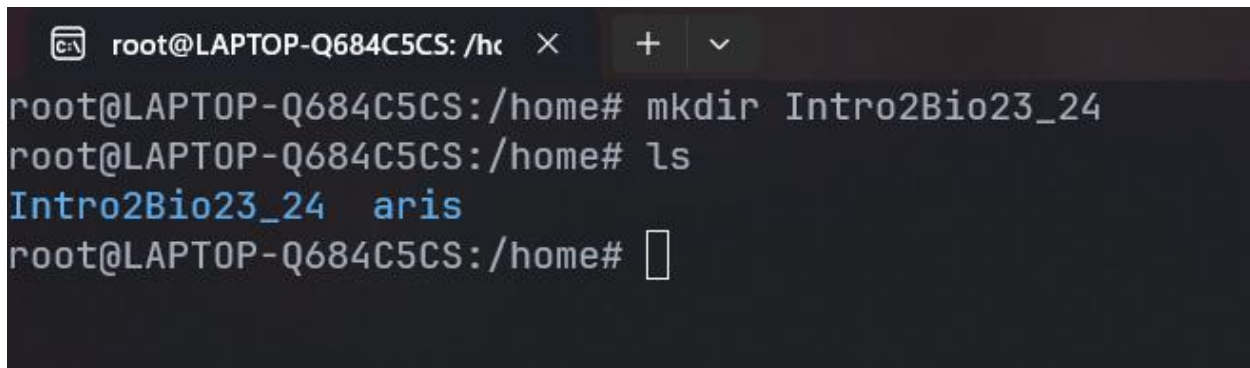
First requirement

Create inside your home folder a new directory named Intro2Bio23_24



```
aris@LAPTOP-Q684C5CS: /ho × + v
aris@LAPTOP-Q684C5CS:~$ pwd
/home/aris
aris@LAPTOP-Q684C5CS:~$ cd ..
aris@LAPTOP-Q684C5CS:/home$ ls
aris
aris@LAPTOP-Q684C5CS:/home$
```

Figure 1.1: Directly after Figure 1, the navigation to the required folders and a look at the contents.



```
root@LAPTOP-Q684C5CS: /hc × + v
root@LAPTOP-Q684C5CS:/home# mkdir Intro2Bio23_24
root@LAPTOP-Q684C5CS:/home# ls
Intro2Bio23_24 aris
root@LAPTOP-Q684C5CS:/home#
```

Figure 1.2: The command “sudo -s” was run directly before the contents of the image. For clarification look at the notes at [\[A priori setup\]](#). This concludes requirement #1.

Second requirement

Inside the folder Intro2Bio23_24 create two new folders named 2023 and 2024



```
root@LAPT0P-Q684C5CS:/home# cd Intro2Bio23_24/
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24# ls
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24# mkdir 2023
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24# mkdir 2024
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24# ls
2023  2024
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 2.1: Completion of requirement #2.

Third requirement

Delete the folder Intro2Bio23_24

```
root@LAPT0P-Q684C5CS:/home/Intro2Bio23_24# cd ..
root@LAPT0P-Q684C5CS:/home# ls
Intro2Bio23_24  aris
root@LAPT0P-Q684C5CS:/home# rm -rf ./Intro2Bio23_24/
root@LAPT0P-Q684C5CS:/home# ls
aris
root@LAPT0P-Q684C5CS:/home#
```

Figure 3.1: Completion of requirement #3. Instead of manually removing the subdirectories and files of the folder a recursive force was preferred.

Fourth requirement

Repeat step 1

To avoid having to type out the commands again, we look at the history in the terminal and go back up to the appropriate command.



```
root@LAPTOP-Q684C5CS:/home# history
 1  ls
 2  rmdir ./Intro2Bio23_24/
 3  ls
 4  clear
 5  neofetch
 6  neo fetch
 7  clear
 8  mkdir Intro2Bio23_24
 9  ls
10  cd Intro2Bio23_24/
11  ls
12  mkdir 2023
13  mkdir 2024
14  ls
15  cd ..
16  ls
17  rm -rf ./Intro2Bio23_24/
18  ls
19  hist
20  history
root@LAPTOP-Q684C5CS:/home# mkdir Intro2Bio23_24
root@LAPTOP-Q684C5CS:/home# ls
Intro2Bio23_24 aris
root@LAPTOP-Q684C5CS:/home#
```

Figure 4.1: Completion of requirement #4.

Fifth requirement

Inside folder Intro2Bio23_24 using the command echo create a file named myname.txt, containing your first name at the first line and your last name at its second and last line

To complete this requirement, the manual page was read to find the appropriate flag for the “echo” command.



```
root@LAPTOP-Q684C5CS: /home# man echo
ECHO(1)
NAME
    echo - display a line of text
SYNOPSIS
    echo [SHORT-OPTION]... [STRING]...
    echo LONG-OPTION
DESCRIPTION
    Echo the STRING(s) to standard output.
    -n    do not output the trailing newline
    -e    enable interpretation of backslash escapes
    -E    disable interpretation of backslash escapes (default)
    --help display this help and exit
    --version
           output version information and exit
    If -e is in effect, the following sequences are recognized:
    \\    backslash
    \a    alert (BEL)
    \b    backspace
    \c    produce no further output
    \e    escape
    \f    form feed
    \n    new line
Manual page echo(1) line 1 (press h for help or q to quit)
```

Figure 5.1: Checking the manual page of the “echo” command. The appropriate flag is marked with a red outline.

Upon finding the appropriate flag the commands are as follow.

```
root@LAPTOP-Q684C5CS:/home# ls
Intro2Bio23_24 aris
root@LAPTOP-Q684C5CS:/home# cd Intro2Bio23_24/
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# echo -e "Aris\nPodotas" >myname.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
myname.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat myname.txt
Aris
Podotas
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 5.2: Completion of requirement #5 using the appropriate flags found prior.

Sixth requirement

Print on screen the contents of file myname.txt



```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat myname.txt
Aris
Podotas
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 6.1: Completion of requirement #6.

Seventh requirement

Create a file num5.txt containing 5 lines, where each line consists of the number 1 to 5 respectively

First we check the default output of the “seq” command, upon seeing that it coincides with the requirement we write the output to the appropriate file.

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq 5
1
2
3
4
5
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq 5 > num5.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num5.txt
1
2
3
4
5
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 7.1: Completion of requirement #7.

Eighth requirement

Create a file num100.txt containing 100 lines, where each line consists of the number 1 to 100 respectively



```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq 100 > num100.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num100.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

Figure 8.1: Completion of requirement 8. The rest of the file does not fit in one image but contains the remaining digits.

Nineth requirement

Count the number of characters of the two files num5.txt and num100.txt



```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# wc num5.txt num100.txt
  5   5  10 num5.txt
100 100 292 num100.txt
105 105 302 total
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 9.1: Completion of requirement #9.

Tenth requirement

Print on screen only the lines of the file num100.txt containing the digit 0

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man grep
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep 0 num100.txt
10
20
30
40
50
60
70
80
90
100
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 10.1: Completion of requirement #10.

Eleventh requirement

Print on screen the first 5 lines of the file num100.txt, ordered from the largest to the smallest number

To complete this requirement, a familiarization with the flags of the “sort” command through the manual was needed.



```
root@LAPTOP-Q684C5CS: /hc x + v
SORT(1) User Commands SORT(1)

NAME
    sort - sort lines of text files

SYNOPSIS
    sort [OPTION]... [FILE]...
    sort [OPTION]... --files0-from=F

DESCRIPTION
    Write sorted concatenation of all FILE(s) to standard output.

    With no FILE, or when FILE is -, read standard input.

    Mandatory arguments to long options are mandatory for short options too. Ordering options:

    -b, --ignore-leading-blanks
        ignore leading blanks

    -d, --dictionary-order
        consider only blanks and alphanumeric characters

    -f, --ignore-case
        fold lower case to upper case characters

    -g, --general-numeric-sort
        compare according to general numerical value

    -i, --ignore-nonprinting
        consider only printable characters

    -M, --month-sort
        compare (unknown) < 'JAN' < ... < 'DEC'

    -h, --human-numeric-sort
        compare human readable numbers (e.g., 2K 1G)

    -n, --numeric-sort
```

Manual page sort(1) line 1 (press h for help or q to quit)

Figure 11.1: The manual page of the “sort” command. The appropriate flag is marked in a red outline.



```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# head 5 myname.txt | sort
head: cannot open '5' for reading: No such file or directory
=> myname.txt <=
Aris
Podotas
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# head myname.txt 5 | sort
head: cannot open '5' for reading: No such file or directory
=> myname.txt <=
Aris
Podotas
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# head num100.txt 5 | sort
head: cannot open '5' for reading: No such file or directory
1
10
2
3
4
5
6
7
8
9
=> num100.txt <=
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man sort
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# head num100.txt 5 | sort -n
head: cannot open '5' for reading: No such file or directory
=> num100.txt <=
1
2
3
4
5
6
7
8
9
10
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 11.2: Completion of requirement #11. Trial and error of using the commands properly is shown as well.

Twelfth requirement

Create a file num100_row.txt containing all numbers from 1 to 100 in one line, separated by a semicolon (;)

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man seq
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq -s ; 100 > num100_row.txt
seq: option requires an argument -- 's'
Try 'seq --help' for more information.
100: command not found
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq -s ";" 100 > num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num100_row.txt
1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40;41;42;43;44;45;46;47;48;49;50;51;52;53;54;55;56;57;58;59;60;61;62;63;64;65;66;67;68;69;70;71;72;73;74;75;76;77;78;79;80;81;82;83;84;85;86;87;88;89;90;91;92;93;94;95;96;97;98;99;100
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure12.1: Completion of requirement #12. The manual page of the “seq” command was re-read to write the appropriate command though it is not shown.



Thirteenth requirement

Print on screen the file num100_row.txt, substituting all semicolons with underscores (_)

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num100_row.txt | tr ";" "_"
1_2_3_4_5_6_7_8_9_10_11_12_13_14_15_16_17_18_19_20_21_22_23_24_25_26_27_28_29_30_31_32_33_34_35_36_37_38_39_40_41_42_43_44_45_46_47_48_49_50_51_52_53_54_55_56
_57_58_59_60_61_62_63_64_65_66_67_68_69_70_71_72_73_74_75_76_77_78_79_80_81_82_83_84_85_86_87_88_89_90_91_92_93_94_95_96_97_98_99_100
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```

Figure 13.1: Completion of requirement #13.

Fourteenth requirement

Print on screen from file num100_row.txt only the numbers 44 and 88

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man grep
General Output Control
-c, --count
    Suppress normal output; instead print a count of matching lines for each input file. With the -v, --invert-match option (see below), count non-matching lines.

--color[=WHEN], --colour[=WHEN]
    Surround the matched (non-empty) strings, matching lines, context lines, file names, line numbers, byte offsets, and separators (for fields and groups of context lines) with escape sequences to display them in color on the terminal. The colors are defined by the environment variable GREP_COLORS. The deprecated environment variable GREP_COLOR is still supported, but its setting does not have priority. WHEN is never, always, or auto.

-L, --files-without-match
    Suppress normal output; instead print the name of each input file from which no output would normally have been printed.

-l, --files-with-matches
    Suppress normal output; instead print the name of each input file from which output would normally have been printed. Scanning each input file stops upon first match.

-m NUM, --max-count=NUM
    Stop reading a file after NUM matching lines. If the input is standard input from a regular file, and NUM matching lines are output, grep ensures that the standard input is positioned to just after the last matching line before exiting, regardless of the presence of trailing context lines. This enables a calling process to resume a search. When grep stops after NUM matching lines, it outputs any trailing context lines. When the -c or --count option is also used, grep does not output a count greater than NUM. When the -v or --invert-match option is also used, grep stops after outputting NUM non-matching lines.

-o, --only-matching
    Print only the matched (non-empty) parts of a matching line, with each such part on a separate output line.

-q, --quiet, --silent
    Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the -s or --no-messages option.

-s, --no-messages
    Suppress error messages about nonexistent or unreadable files.

Output Line Prefix Control
-b, --byte-offset
    Manual page grep(1) line 72 (press h for help or q to quit)
```

Figure 14.1: The manual page of the “grep” command. The flag we will be using is highlighted in red.

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep "44|88" num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep '44' '88' num100_row.txt
grep: 88: No such file or directory
num100_row.txt:1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40;41;42;43;44;45;46;47;48;49;50;51
;52;53;54;55;56;57;58;59;60;61;62;63;64;65;66;67;68;69;70;71;72;73;74;75;76;77;78;79;80;81;82;83;84;85;86;87;88;89;90;91;92;93;94;95;96;97;98;99;100
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man grep
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man grep
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep -o '44 | 88' num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep -o '44|88' num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep -o "44|88" num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep -o "44"|"88" num100_row.txt
88: command not found
^C
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep "44|88" num100_row.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man grep
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# grep -o -- '44|88' num100_row.txt
44
88
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24#
```



Figure 14.2: completion of requirement #14. All the trials before the final working command are shown. Notice that the “—” in the final few commands is unnecessary, the only thing different to the final command with the non-working ones is the use of the escape character “\|” instead of a simple pipe “|”.

Fifteenths requirement

Create a file `num_2_cols.txt` containing 2 columns. The first column contains the numbers from 1 to 100, ordered ascending from the smallest to the greatest. The second column contains the numbers from 1 to 100, ordered descending from the greatest to the smallest (hence, the sum of each line should be 101). For the specific task, the paste command might be helpful.

```
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man paste
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
myname.txt  num100.txt  num100_row.txt  num5.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man paste
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq 100 1 | paste num100_row.txt >num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
myname.txt  num100.txt  num100_row.txt  num5.txt  num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num_2cols.txt
1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40;41;42;43;44;45;46;47;48;49;50;51;52;53;54;55;56;57;58;59;60;61;62;63;64;65;66;67;68;69;70;71;72;73;74;75;76;77;78;79;80;81;82;83;84;85;86;87;88;89;90;91;92;93;94;95;96;97;98;99;100
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# vim
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# seq 100 1 >holder.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
holder.txt  myname.txt  num100.txt  num100_row.txt  num5.txt  num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# vim
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man seq
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# rm holder.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
myname.txt  num100.txt  num100_row.txt  num5.txt  num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# sort | paste num100.txt >num_2cols.txt
^C
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# man sort
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# sort -n -r num100.txt | paste num100.txt >num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
myname.txt  num100.txt  num100_row.txt  num5.txt  num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num_2
cat: num_2: No such file or directory
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num_2cols.txt
1
2
3
4
5
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# vim
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# sort -n -r num100.txt >holder.txt | paste num100.txt holder.txt >num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# vim
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# ls
holder.txt  myname.txt  num100.txt  num100_row.txt  num5.txt  num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat holder
cat: holder: No such file or directory
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat holder.txt
100
99
98
97
96
95
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# paste num100.txt holder.txt > num_2cols.txt
root@LAPTOP-Q684C5CS:/home/Intro2Bio23_24# cat num_2cols.txt
```

Figures 15.1, 15.2, 15.3: Solution to the final requirement. All attempts are shown.



Explanation of solution:

The paste command will be used to take two inputs (the numbered files or variables) and make one, the final output. To do this the manual was read.

All the solutions use the num100.txt for one of the inputs to the paste command since it is the file that contains the data for the first of the two columns of the output file.

This means that up to this point the final command looks like “* | paste num100.txt * > num_2cols.txt” and the question is to fill the “*” with the appropriate commands.

First attempt

Using seq to make the reverse list. This did not work for a few reasons; the wrong file was used (num100_rows.txt instead of num100.txt) and the syntax for the seq command does not work the way envisioned (seq 100 1 does not work).

Second attempt

Use the sort command to write the inverse of num100.txt into a file and use that in the paste command. This does not work in one line with the “|” operator so the command was split into two.