**Fun with scripting tools**

A screenshot of a computer

Description automatically generatedI navigated to the 3320 folder to find the 4HKD.pdb file and I made a copy into my home directory.

After getting the 4HKD.pdb file, I used grep to generate the header by using the command:

“grep -Ev '^(ATOM|CONECT|HETATM|TER|END)' 4HKD.pdb” and got the result.

A screenshot of a cell phone

Description automatically generatedA screenshot of a computer

Description automatically generated

For number 2, I used sed to change the records that have “HETATM” and “MSE” into “ATOM using the command:

sed -e 's/HETATM/ATOM/g' -e 's/MSE/ATOM/g' 4HKD.pdb > temp.txt

The corrected lines:

A screenshot of a computer

Description automatically generated

For number 3, I used awk to find the maximum and minimum values for the ATOMs and separated into the x y and z. For the maximum I used the command:

printf "x\n" && grep ^'ATOM' 4HKD.pdb | sort -nk7 | head -1 | awk 'BEGIN {max = -999999} {if ($7>max) max=$7} END {printf max}' && printf "\ny\n" && grep ^'ATOM' 4HKD.pdb | sort -nk8 | head -1 | awk 'BEGIN {max = -999999} {if ($8>max) max=$8} END {printf max}' && printf "\nz\n" && grep ^'ATOM' 4HKD.pdb | sort -nk9 | head -1 | awk 'BEGIN {max = -999999} {if ($9>max) max=$9} END {printf max}' && printf "\n"

A screenshot of a cell phone

Description automatically generated

For the minimum I used the command:

printf "\n X:" && grep ^'ATOM' 4HKD.pdb | sort -nk7 | head -1 | awk 'BEGIN {min = 1000000} {if ($7<min) min=$7} END {printf min}' && printf "\n Y:" && grep ^'ATOM' 4HKD.pdb | sort -nk8 | head -1 | awk 'BEGIN {min = 1000000} {if ($8<min) min=$8} END {printf min}' && printf "\n Z:" && grep ^'ATOM' 4HKD.pdb | sort -nk9 | head -1 | awk 'BEGIN {min = 1000000} {if ($9<min) min=$9} END {printf min}' && printf "\n"

A screenshot of a cell phone

Description automatically generated

For number 4, I also used awk with grep to sort and find the mean values for the HETATM records. I used the command:

printf "x\n" && grep ^'HETATM' 4HKD.pdb | sort -nk7 | head -1 | awk 'BEGIN {mean = 0; count = 0} {mean = mean + $7; count = count + 1} END {mean = mean/count; printf mean}' && printf "\n" && printf "y\n" && grep ^'HETATM' 4HKD.pdb | sort -nk8 | head -1 | awk 'BEGIN {mean = 0; count = 0} {mean = mean + $8; count = count + 1} END {mean = mean/count; printf mean}' && printf "\n" && printf "z\n" && grep ^'HETATM' 4HKD.pdb | sort -nk9 | head -1 | awk 'BEGIN {mean = 0; count = 0} {mean = mean + $9; count = count + 1} END {mean = mean/count; printf mean}' && printf "\n"

A screenshot of a computer

Description automatically generated

For Number 5, I used the command: "sed -e 's/HOH/WAT/g' 4HKD.pdb > question5.txt" to change all the 'HOH' into 'WAT' using sed.



A screenshot of a computer

Description automatically generatedchanged txt file:

For number 6, I used the grep command: "grep ^'ATOM' 4HKD.pdb | sort -nk11" to sort the 11th column of all ATOM records.

A screenshot of a computer

Description automatically generated

**Lab Work:**

1. I created a lab2data.txt file and used nano to generate 6 lines of using random numbers.

A screenshot of a computer

Description automatically generated

1. I created an awk file that converted the string of the form +- n/m/o into an absolute number of knuts.

A screenshot of a cell phone

Description automatically generatedA screenshot of a computer

Description automatically generated

1. A screenshot of a cell phone

   Description automatically generatedSaved the output to work on the next part.

A screenshot of a computer

Description automatically generated

I created a second awk command named knuts2string.sh that did the inverse of the first awk command. The command: awk '{knuts = $1 % 17}{sickles = $1/ 17}{galleons = int(sickles) / 23} {sickles = int(sickles) % 23}{printf int(galleons)}{printf "/"}{printf int(sickles)}{printf "/"}{printf int(knuts)}{printf "\n"}{printf "\n"}' tmp.txt

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedT**hen I checked the output by using the tmp.txt file and it matched the original file that was in the lab2data.txt**

For the last part I used the command: cat lab2data.txt | awk 'BEGIN{sum = 0}{sum = sum + $1}END{printf sum}' tmp.txt >> lab2data.txt added all the numbers in the tmp.txt file and outputted the sum into lab2data.txt

A screenshot of a computer

Description automatically generatedA screenshot of a cell phone

Description automatically generated