Linux:

1. What does ls -alt do?

Lists most recent modified files

2. What command would you use to list all files starting with 'Run' and ending with '.txt' in a directory and all of its subdirectories?

ls Run*.txt

3. How would you append the contents of 'exampleFile1.txt' to 'exampleFile2.txt'?

Cat exampleFile1.txt > exampleFile2.txt

4. How would you (1) sort the contents of 'exampleFile1' and (2) redirect the sorted content to 'exampleFile2.txt' in one line using the pipe operator?

cat 'exampleFile1' | sort > exampleFile2.txt

5. Which commands would you use to find files whose name match a certain pattern, and to find files containing a certain text?

find , grep

SQL: 1. For the following SQL statement, what is wrong with it and how would you fix it:

-- Question:

SELECT UserId, AVG(Total) AS AvgOrderTotal FROM Invoices

groupby(userid)

HAVING COUNT(OrderId) >= 1

Bioinformatics:

• Recursively find all FASTQ files in a directory and report each file name and the percent of sequences in that file that are greater than 30 nucleotides long.

```
#!/bin/bash

echo File Percentage
for file in *.fastq
do
        num_of_lines=$(cat $file | wc -l)
        num_of_lines=$((num_of_lines/4))
        num_of_lines_seq30=$(cat $file | awk '(NR%4==2)' | awk 'length($1)  >= 30' | wc -l)
        percentage=$(echo $num_of_lines_seq30 \/ $num_of_lines |bc -l)
        percentage=$(expr $percentage*100 | bc)
        echo $file $percentage
done
```

• Given a FASTA file with DNA sequences, find 10 most frequent sequences and return the sequence and their counts in the file.

```
list_headers = []
list_seqs = []
currentSeq = ''

fh = open("/Users/kadam/Desktop/sample.fasta","r")

for line in fh:
    if line.startswith(">"):
        line = line.strip()
        list_headers.append(line)

        if currentSeq != '':
            list_seqs.append(currentSeq)
            currentSeq = ''

#concatenates the lines by stripping the end of line character
    else:
```

```
        currentSeq += line.strip()

SeqCount = {}

for i in list_seqs:
    if i in SeqCount:
        SeqCount[i] +=1
    else:
        SeqCount[i] =1

sorted(SeqCount, key=SeqCount.get, reverse=True)[:10]
```

• Given a chromosome and coordinates, write a program for looking up its annotation. Keep in mind you'll be doing this annotation millions of times.

```
        import pandas as pd

        ch=input("Enter Chromosome: ")
        coor=input("Enter start coordinate: ")
        coor=int(coor)

        gene_data = pd.read_csv("/Users/kadam/Desktop/hg19_annotations.gtf", sep='\t',
                names=["Chromosome","refFlat","Region","start_Coordinate","end_Coordinate","a","b","c","d","e"])
        gene_data["Info"] = gene_data["a"] + gene_data["b"] + gene_data["c"] + gene_data["d"]
        gene_data = gene_data.drop(['a','b','c','d','e'], axis = 1)

        reduced_data = gene_data[gene_data['Chromosome'].str.match(ch)]
        reduced_data.loc[reduced_data['start_Coordinate'] == coor]
```

- sample_files.zip
- Tab-delimited file: Chr<tab>Position
- GTF formatted file with genome annotations.

NOTE:

1. Keep in mind; we will use the results of these tasks to assess your ability. This is a chance for you to show off your programming skills and style. 2. A Python solution is ideal, as our code-base is primarily in Python. 3. Sample input files have been provided for each task. 4. Make sure you understand the file

formats (FASTQ, FASTA, GTF) to perform these tasks correctly. 5. Please make sure each task can run on the command line. 6. In the spirit of assessing your programming abilities, please avoid using 3rd-party tools to solve these problems (parsers and formatters excluded).

2. Create a repository on GitHub and upload your code there. Make some minor changes to your code locally, and use a local Git installation to commit the changes to your GitHub repository.