Question 1: *Look at the result of this count query. Note that it does not include any name, email, or other personally identifiable information. What can you nevertheless learn about the TA's musical tastes? What possible genres might they have chosen? Alternatively, what genres is it impossible for them to have chosen?*

Answer 1: Based on the result query, the musical taste choices amongst the users are Hip Pop, Pop, Country, Rock, Classical, House, Metal or something else (Prefer not to Answer). Since they are not age 0, we can ignore those rows. Since the TA is a PhD student, he is likely to be 24 years or older. Based on this information, the TA might like Country, Rock, House, Pop, Metal or something else. Hence, it is unlikely that he likes Classical or Hip Hop.

Question 2: *What did you find out about the TA? Are your findings consistent with Question 1? Combine the two together to learn the TA's exact age.*

Answer 2: Based on the LinkedIn profile, Kinan says that he goes to Heavy Metal concerts. Also, it looks like he graduated with a Bachelor's degree in 2015. Assuming that I was right about his age range, it seems that there is only one row that matches both of these plausible facts – Kinan's age is 27, and his favorite music genre is Metal. My findings are consistent with Question 1 because Metal was one of the shortlisted options I gathered.

Question 3: *Identify the TA's favorite color. What is it? How easy or obvious is this to do, and why?*

Answer 3: The TA's favorite color is Black. It was easy to find because I know his exact age, i.e. 27. Furthermore, there is only one count for that age. Hence, I was very certain that it was referring to Kinan's favorite color.

Question 4: *What information can you learn about the TA's favorite sport from the above query?*

Answer 4: Assuming that Kinan mentions his age is 27, his favorite sport is one of the following: Baseball, E-Sports, Hockey.

Question 5: *What is our TA's favorite sport?*

Answer 5: Kinan's age last year was 26. The options from last year showed that ages 25-30 liked American Football, Basketball, Soccer or E-Sports. Assuming that the TA specified his age in both the surveys, the common answer is E-Sports. Hence, our TA's favorite sport is E-Sports.

Question 6: *Run `dp.py` several times varying the epsilon privacy parameter for different values between 10 and 0.01, like so:*

```
$ python3 dp.py 0.01
[...]
$ python3 dp.py 0.1
[...]
[etc.]
```

What happens when the privacy parameter grows larger or smaller? How does that affect privacy?

Answer 6: From the output values, I can see that as the privacy parameter grows smaller, the number of count values that are different from the actual count value increases. This implies that as the privacy parameter decreases, it is harder for the attacker to find the true count value for each group, and hence, determine the actual favorite music of an individual in the survey. In other words, as the privacy parameter grows larger, the privacy of the surveyed individuals decreases further.

Question 7: *Look at the plot generated with privacy parameter epsilon = 0.5. What is the most likely value? What is the expected (i.e., average) value? How do they relate to the actual value (i.e., the query executed without any noise via `client.py`)? How does the plot change for different values of the privacy parameter?*

Answer 7: Based on the plot (dp-plot.png), the most likely value is 1. The expected value is around 0.9, which I would round to 1. The value is equal to the actual count value. As the epsilon value increases, the chances of the generated count being the actual count is a lot more frequent. When I plotted the graph with epsilon = 10 (dp-plot-ep-10.png), there was hardly any frequency for count values apart from 1. Hence, as the privacy parameter increases, the value displayed is more frequently going to be the actual value.

Estimated Average calculation for epsilon = 0.5:
(-6 * 0.015) + (-5 * 0.02) + (-4 * 0.025) + (-3 * 0.0375) + (-2 * 0.075) + (-1 * 0.125) + (0 * 0.175) + (1 * 0.225) + (2 * 0.13) + (3 * 0.08) + (4 * 0.075) + (5 * 0.02) + (6 * 0.02) + (11 * 0.015) + (12 * 0.015)

Question 8: *Run the composition attack against the average age grouped by programming experience. What can you deduce from the exposed averages about the programming experience level of our TA? How confident are you in what you have deduced? Are there scenarios where they might be wrong?*

Answer 8:
When I ran the composition.py script for average age, the following table was printed:

Programming	AVG (age)
1-2 Years	22
2-3 Years	22
3-5 Years	23
5-8 Years	23
8-10 Years	21
Less than one year	22
More than 10 Years	31

Based on these results and Kinan's age, I would think that his programming experience might be 3-5 years, 5-8 years, or more than 10 years. If I have to choose one, it would be more than 10 years. I chose groups that have higher average age because the TA could increase the average age value, especially if he were in a group with few 20, 21, or 22 year old individuals.

However, my confidence is low because he could technically fit any group depending on the number of members in each group. For instance, if there is a group with Kinan and six 20 year old individuals, the mean would be 21 (the lowest number of all groups). Hence, he could statistically be in any group, which makes me feel less confident about my educated guess.

Question 9: *Reuse your composition attack from question 8 to compute the exact non-noised counts per programming experience level. Deduce the programming experience level of our TA, with high confidence, by looking at both the exposed counts and the previously exposed averages. Now summarize everything you've learned about the TA!*

Based on the values I got (different from the table in Question 8), it cannot be more than 10 years because the number of people in that was 2 but the average age was 30. Hence, the ages in that group are 26 and 34. It cannot be less than one year because the average age is 23 and the count is 1. This implies that the age of the individual in that group is 23. It cannot be 8-10 years either because the average age is 21 and the count is 1. Hence, the actual age would be 21. It cannot be 1-2 years because the average age is 22 and there are only 2 people.

Out of the remaining groups, Kinan could either be in 2-3 years, 3-5 years or 5-8 years. He most likely has 5-8 years of programming experience since that is the group with fewer, but higher average age values than the remaining groups' values.

To summarize, the TA is 27 years old and likes Black color. His most favorite music genre is Metal and his most favorite sport is E-Sports. Lastly, he probably has programming experience of about 5-8 years.

Question 10: *Does the class you implemented suffice to truly enforce that a dataset is never used beyond a certain privacy budget? Can developers intentionally or unintentionally over-use the dataset beyond the privacy budget? At a very high level, how would you design a different privacy budget enforcement mechanism that does not suffer these drawbacks?*

No, it does not truly enforce that a dataset is never used beyond a certain privacy budget. Developers can intentionally over-use the dataset by setting a low epsilon value. Hence, one way to enforce privacy budget mechanism is to pre-set epsilons along with each query type. This would also allow us to account for the degree of disclosure for each query type. For instance, a query about average gives out more information than a query about count. Hence, the average query's epsilon value should be higher than the count's.