

1. What is incorrect regarding the LINDDUN threat modelling?
  - ☐ It can be used to identify privacy threats in a software system
  - ☐ A 'wrong password' error message displayed in a login attempt proves the existence of an user account. This can lead to a "Detecting" threat
  - ☐ It is initially performed during software testing
  - ☐ Personal data flow needs to be understood before performing it
  - ☐ Data disclosure threat can lead to non-compliance threat
2. Which of the following is not a key factor when selecting a PET for software applications?
  - ☐ Does the application handle personal data?
  - ☐ Are the resources sufficient to integrate a certain PET?
  - ☐ Is the PET scalable?
  - ☐ Does the application need to maintain the accuracy of the personal data?
  - ☐ Does the PET use cutting-edge technology?
3. A developer used the following Mondrian algorithm code to partition a dataset called data\_anon. What is/are the correct conclusion/s that can be taken by only looking at the given code? Assume that the code and the associated helper functions qid\_variability and split are working correctly without any errors. The functionality of the two helper functions is similar to the qid\_variability and split functions taught during the course.

```
def mondrian():
    final_partitions = []
    partitions = [data_anon.index]

    while partitions:
        partition = partitions.pop(0)

        if len(partition) >= 3 * 2:
            # For a given partition of row indexes get_qid_variability returns a
            # list of QIDs sorted in descending order based on their variability
            qid_variability = get_qid_variability(partition)

            for column in qid_variability:
                # split function correctly splits each partition into 2 which are lhs,rhs
                # Both lhs and rhs will store row indexes
                lhs,rhs = split(partition, column)

                if len(lhs) < 3 or len(rhs) < 3:
                    continue
                else:
                    partitions.extend((lhs,rhs))
                    break
            else:
                final_partitions.append(partition)
        else:
            final_partitions.append(partition)

    return final_partitions
```

- X. Each partition in the final\_partitions will have at least 3 row indexes in it
- Y. The developer achieves 3-anonymity
- Z. Some partitions in the final\_partitions might have more than 3 row indexes

- ☐ X and Y
- ☐ X and Z
- ☐ Y and Z
- ☐ None, or only one of X, Y, and Z
- ☐ All

4. Query: "What is the total number of people in the dataset?"

The real answer of this query is 1000. Differential private results are generated for this query using epsilon values 0.1, 0.01, and 0.001. Which epsilon has the highest probability of generating the noisy result 1000.17?

- ☐ 0.01
- ☐ Cannot decide as the noise is generated randomly
- ☐ 0.1
- ☐ 0.001
- ☐ All epsilons have the same probability