Python

Question 1:

You have an input dictionary given,

input_dict = {"abc":{"def":{"ghi":{"jkl":{"mno":{"pqr":{"stu":{"vwx":{"yz":"you are finally here !!!"}}}}}}}}}

Task: You have to write a Python function that will take this input and print it like that,

output = {"abc":["def","ghi","jkl","mno","pqr","stu","vwx","yz"],
"def":["ghi","jkl","mno","pqr","stu","vwx","yz"],
"ghi":["jkl","mno","pqr","stu","vwx","yz"],
"jkl":["mno","pqr","stu","vwx","yz"],
"mno":["pqr","stu","vwx","yz"],
"pqr":["stu","vwx","yz"],
"stu":["vwx","yz"],
"vwx":["yz"],
"yz":["you are finally here !!!"]}

Answer 1 : Code

Github 1 : https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Python/Question_1.py

Question 2:

Given an array of length ‘N’, where each element denotes the position of a stall. Now you have ‘N’ stalls and an integer ‘K’ which denotes the number of horses that are mad. To prevent the horses from hurting each other, you need to assign the horses to the stalls, such that the minimum distance between any two of them is as large as possible. Return the largest minimum distance.

Answer 2 :

This is a question of large K largest distance also. In which we need to find the largest distance between the given inputs and the K value.

Github 2 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Python/Question_2.py

Question 3 :

Mr. Karthiken works in a door mat manufacturing company. One day, he designed a new door mat with the following specifications:

a) Mat size must be N X M. (N is an odd natural number, and M is 3 times N.)
b) The design should have 'WELCOME' written in the center.
c) The design pattern should only use |, . and – characters.

Sample Design is given above image, Write a python code for this.

Answer 3 :

This is question named Designer Door Mat where we design it with NxM in size
Here N should represent the Width of Mat and M should represent the length of Mat

Github 3 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Python/Question_3.py

Question 4 :

Given an array nums of n integers, return an array of all the unique quadruplets [nums[a], nums[b], nums[c], nums[d]] such that:

a) 0 <= a, b, c, d < n
b) a, b, c, and d are distinct.
c) nums[a] + nums[b] + nums[c] + nums[d] == target

Answer 4 :

To hold unique quadruplets, this code makes use of a set. It uses two nested loops to cycle through all possible pairings of elements after sorting the input array nums. It uses a two-pointer method inside the nested loops to locate the additional two items so that the total

matches the desired amount. By storing the quadruplets in a set, it guarantees uniqueness. Ultimately, the list of distinct quadruplets is returned.

Github 4 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Python/Question_4.py

SQL

Question 1:

Given the following tables:

```
sql> SELECT * FROM runners;
+----+--------------+
| id | name         |
+----+--------------+
|  1 | John Doe     |
|  2 | Jane Doe     |
|  3 | Alice Jones  |
|  4 | Bobby Louis  |
|  5 | Lisa Romero  |
+----+--------------+

sql> SELECT * FROM races;
+----+----------------+-----------+
| id | event          | winner_id |
+----+----------------+-----------+
|  1 | 100 meter dash |  2        |
|  2 | 500 meter dash |  3        |
|  3 | cross-country  |  2        |
|  4 | triathalon     |  NULL     |
+----+----------------+-----------+
```

What will be the result of the query below?
SELECT * FROM runners WHERE id NOT IN (SELECT winner_id FROM races)
Explain your answer and also provide an alternative version of this query that will avoid the issue that it exposes.

Answer 1 :

All runners who are not identified as winners in the races table will be returned by the query. But there's a problem with the way it treats NULL values.

Explanation
Subquery: To select every winner_id value in the races table, use the subquery SELECT winner_id FROM races.
NOT IN: The NOT IN operator determines whether the list of winner_id values obtained from the subquery contains the id of a runner from the runners table.
NULL Values: The issue is that in races where a winner hasn't been declared yet, winner_id may be NULL in the races table. The NOT IN operator handles NULL in a unique way. An identifier for a runner that is absent from the list—including NULL values—will be regarded as a match for NOT IN.
Outcome:

The result will include runners whose id values are missing from the winner_id list (including those whose winner_id is NULL).
The results will not include runners whose id values match the current winner_id values (except from NULL).

The result will include runners whose id values are missing from the winner_id list (including those whose winner_id is NULL).
The results will not include runners whose id values match the current winner_id values (except from NULL).

Alternative Query (using LEFT JOIN): To address the NULL value issue and ensure we only select runners who haven't won any races, we can use a LEFT JOIN with an additional filtering condition:

```
SELECT r.id, r.name
FROM runners r
LEFT JOIN races rc ON r.id = rc.winner_id
WHERE rc.winner_id IS NULL;
```

Justification

LEFT JOIN: Under the condition r.id = rc.winner_id, this joins the runners table (r) with the races table (rc).
Outer Join: Even if there isn't a match in the right table (races), the LEFT JOIN incorporates all rows from the left table (runners).
Clause WHERE: We only choose runners from the runners table when the matching winner_id in the races table is NULL, because of the WHERE rc.winner_id IS NULL condition. This ensures that none of these runners have ever won a race.
Whether a winner has been declared for a given race, or if there are races with NULL winner_id, this alternate query efficiently detects runners who are not listed as winners in the races table.

Question 2 :

Given two tables created as follows

```
create table test_a(id numeric);

create table test_b(id numeric);

insert into test_a(id) values
  (10),
  (20),
  (30),
  (40),
  (50);

insert into test_b(id) values
  (10),
  (30),
  (50);
```

Write a query to fetch values in table test_a that are and not in test_b without using the NOT keyword.

Answer 2 :

SELECT a.* FROM test_a a LEFT JOIN test_b b ON a.id = b.id WHERE b.id IS NULL;

We came to this outcome by performing a left join and filtering out rows where the join produces a match.

We can also use the EXCEPT keyword.

Question 3 :

Given the following tables:

```
SELECT * FROM users;

user_id  username
1        John Doe
2        Jane Don
3        Alice Jones
4        Lisa Romero

SELECT * FROM training_details;

user_training_id  user_id  training_id  training_date
1                 1        1            "2015-08-02"
2                 2        1            "2015-08-03"
3                 3        2            "2015-08-02"
4                 4        2            "2015-08-04"
5                 2        2            "2015-08-03"
6                 1        1            "2015-08-02"
7                 3        2            "2015-08-04"
8                 4        3            "2015-08-03"
9                 1        4            "2015-08-03"
10                3        1            "2015-08-02"
11                4        2            "2015-08-04"
12                3        2            "2015-08-02"
13                1        1            "2015-08-02"
14                4        3            "2015-08-03"
```

Write a query to get the list of users who took a training lesson more than once in the same day, grouped by user and training lesson, each ordered from the most recent lesson date to oldest date.

Answer 3 :

```
SELECT u.user_id, username, training_id, training_date
FROM users u
INNER JOIN
( SELECT user_id, training_id, training_date, COUNT(*) AS count
FROM training_details
GROUP BY user_id, training_id, training_date
HAVING COUNT(*) > 1 )
AS t ON u.user_id = t.user_id
ORDER BY t.user_id, t.training_id, t.training_date DESC;
```

Question 4 :

Consider the Employee table below.

Emp_Id	Emp_name	Salary	Manager_Id
10	Anil	50000	18
11	Vikas	75000	16
12	Nisha	40000	18
13	Nidhi	60000	17
14	Priya	80000	18
15	Mohit	45000	18
16	Rajesh	90000	–
17	Raman	55000	16
18	Santosh	65000	17

Write a query to generate below output:

Manager_Id	Manager	Average_Salary_Under_Manager
16	Rajesh	65000
17	Raman	62500
18	Santosh	53750

Answer 4 :

```
SELECT e2.emp_id, e2.emp_name, AVG(e1.salary) AS avg_salary
FROM manager_emp e1
INNER JOIN manager_emp AS e2
ON e1.manager_id = e2.emp_id
GROUP BY e2.emp_id, e2.emp_name;
```

Statistics:

Question 1 :

What is the meaning of six sigma in statistics? Give proper example

Answer 1 :

As a data scientist, Six Sigma is a methodology focused on reducing errors and variations in processes, aiming for exceptional outcomes by minimizing the likelihood of defects. It sets a stringent quality standard of 3.4 faults per million opportunities, striving for near-perfection. In a manufacturing context, implementing Six Sigma involves defining faults, measuring defect rates, analyzing root causes, enhancing processes, controlling quality, and verifying improvements. Achieving Six Sigma level quality leads to significant reductions in defects, increased customer satisfaction, and improved profitability.

Question 2 :

What type of data does not have a log-normal distribution or a Gaussian distribution? Give proper example

Answer 2 :

Non-normally distributed data encompass a range of patterns that deviate from the typical bell-shaped curve seen in Gaussian or log-normal distributions. Examples include skewed data, with most values clustered at one end and a tail extending towards the other. Bimodal or multimodal data exhibit multiple peaks, indicating distinct clusters with different central tendencies. Heavy-tailed distributions have a higher likelihood of extreme values, while discrete distributions involve specific, separate data points. Understanding these diverse patterns is crucial for selecting appropriate statistical methods and interpreting results effectively in data analysis.

Question 3 :

What is the meaning of the five-number summary in Statistics? Give proper example

Answer 3 :

In statistics, the five-number summary provides a concise summary of the central tendency and spread of a dataset. It consists of five key values: the minimum, first quartile (Q1), median

(second quartile or Q2), third quartile (Q3), and maximum. These values help understand the distribution and variability of the data without needing to examine the entire dataset.

For example, consider the following dataset of exam scores: [65, 70, 75, 80, 85, 90, 95, 100]. The five-number summary would be:
Minimum: 65
Q1 (First Quartile): 72.5
Median (Second Quartile): 82.5
Q3 (Third Quartile): 92.5
Maximum: 100

This summary indicates that the lowest score is 65, the highest is 100, and the middle value (median) is 82.5. Additionally, it shows that 25% of the data falls below 72.5, and 75% falls below 92.5, providing insights into the distribution of the exam scores.

Question 4 :

What is correlation? Give an example with a dataset & graphical representation on jupyter Notebook

Answer 4 :

Correlation is a statistical measure that quantifies the strength and direction of the relationship between two variables. It indicates how closely the movements of one variable are associated with the movements of another. Positive correlation means that as one variable increases, the other tends to increase as well, while negative correlation indicates that as one variable increases, the other tends to decrease. The correlation coefficient, typically denoted by "r," ranges from -1 to 1, with values closer to -1 or 1 indicating stronger correlation, and values closer to 0 indicating weaker correlation or no correlation at all.

Github 4 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Statistics/Question4.ipynb

Machine learning :

Question 1 : In program
Answer 1 : In program

Github 1 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Machine_Learning/Question1.ipynb

Question 2 : In program
Answer 2 : In program
Github 2 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Machine_Learning/Question2.ipynb

Question 3 : In program
Answer 3 : In program
Github 3 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Machine_Learning/Question3.ipynb

Deep Learning :

Question 1 :

(a) Explain how you can implement DL in a real-world application.
(b) What is the use of Activation function in Artificial Neural Networks? What would be the problem if we don't use it in ANN networks.

Answer 1 :

(a)
As a data scientist, implementing deep learning (DL) in a real-world application involves these concise steps:

1. Define objectives: Clearly outline the problem you aim to solve or the task you want to accomplish using DL.
2. Data acquisition: Gather relevant data from various sources, ensuring it's of high quality and sufficient for training a DL model.

3. Data preprocessing: Clean, preprocess, and format the data to make it suitable for DL training. This may include tasks like normalization, encoding categorical variables, and handling missing values.
4. Model selection: Choose an appropriate DL model architecture based on the nature of the problem and data characteristics. Common architectures include convolutional neural networks (CNNs) for image tasks, recurrent neural networks (RNNs) for sequential data, and transformer models for natural language processing (NLP) tasks.
5. Model training: Train the selected DL model using the preprocessed data. Optimize hyperparameters, adjust model architecture if needed, and monitor training progress using validation metrics.
6. Model evaluation: Assess the trained model's performance using evaluation metrics and validation datasets to ensure it meets predefined criteria for accuracy, precision, recall, etc.
7. Deployment: Deploy the trained DL model into a production environment, integrating it with existing systems or applications to make predictions on new, unseen data.
8. Monitoring and maintenance: Continuously monitor the deployed DL model's performance in production, monitoring metrics such as accuracy, latency, and resource usage. Retrain the model periodically using new data to maintain its effectiveness and adapt to changing conditions.

   By following these steps, a data scientist can successfully implement DL in a real-world application, leveraging its capabilities to solve complex problems and derive valuable insights from data.

(b)

Activation functions are crucial components in artificial neural networks (ANNs) as they introduce non-linearity into the model, allowing it to learn complex patterns and relationships within the data. Without activation functions, ANNs would essentially reduce to linear transformations, rendering them incapable of capturing the intricate features and nuances present in real-world data. By introducing non-linearity, activation functions enable ANNs to model highly nonlinear relationships, making them powerful tools for various tasks such as classification, regression, and pattern recognition. Therefore, activation functions

play a vital role in enhancing the expressive power and effectiveness of artificial neural networks in solving complex real-world problems.

Question 2 :

Train a Pure ANN with less than 10000 trainable parameters using the MNIST Dataset

Answer 2 : In program
Github 2 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Deep_Learning/Question_2.py

Question 3 :

Perform Regression Task using ANN
Note: You are feel free to use any Regression ML dataset

Answer 3 : In program
Github 3 :
https://github.com/RishavMishraRM/Ineuron_Assesment/blob/main/Deep_Learning/Question_3.py