cognizant®

Exam Pattern for Genc Cluster 1, 2 and 3

Cluster 1

2 SQL Queries -35 Mins
2 Coding Questions of Java -60 Mins
1 Web Page making (HTML, CSS & JavaScript)-25 Mins

Cluster 2

2 SQL Queries -35 Mins
2 Coding Questions of Python -70 Mins
10 Questions from Cloud Fundamentals -15 Mins

Cluster 3

2 SQL Queries -35 Mins
2 Coding Questions of C# -60 Mins
1 Web Page making (HTML, CSS & JavaScript)-25 Mins

cognizant®

Cluster 1

Technical Assessment Preparation:

- Focus on Java and ANSI SQL (85% of the questions are based on these).
- Practice the following coding problems:
  - Convert Roman numerals to decimal numbers (1 to 3999).
  - Count the number of substrings in a binary string that start and end with 1.
  - Solve number series problems like:
    - Finding the sum of all prime numbers between 1 and N.
    - Identifying prime numbers from 1 to N.

Cluster 1 Coding Questions

1. Prime Numbers from 1 to N: Given a number N, the task is to find the Prime Numbers from 1 to N.
2. Largest Element in Array: Find the largest element in an array.
3. Second Largest Element in Array: Find the second largest element in an array.
4. Sum of Elements in an Array: Write a program to find the sum of elements in a given array.
5. Element with Left Side Smaller and Right Side Greater: Find an element in an array such that all elements to its left are smaller and all elements to its right are greater.
6. Missing Number in Array: Find the missing number in an array.
7. Counting Elements in Two Arrays: Count the number of elements in two arrays that are equal.
8. Elements in the Range: Find elements in a given range in a sorted array.
9. Last Duplicate Element in a Sorted Array: Find the last duplicate element in a sorted array.
10. Program for Average of an Array: Write a program to find the average of elements in an array.

Cluster 1 Coding Questions

- **Implement Bubble Sort**: Write a function to sort an array using the Bubble Sort algorithm.
- **Implement Insertion Sort**: Write a function to sort an array using the Insertion Sort algorithm.
- **Implement Selection Sort**: Write a function to sort an array using the Selection Sort algorithm.
- **Implement Merge Sort**: Write a function to sort an array using the Merge Sort algorithm.
- **Implement Quick Sort**: Write a function to sort an array using the Quick Sort algorithm.
- **Implement Heap Sort**: Write a function to sort an array using the Heap Sort algorithm.
- **Sort a Nearly Sorted Array**: Given an array where each element is at most k positions away from its target position, write a function to sort the array.
- **Sort an Array of 0s, 1s, and 2s**: Write a function to sort an array consisting of only 0s, 1s, and 2s.
- **Sort an Array with Repeated Elements**: Write a function to sort an array that contains duplicate elements.
- **Sort an Array with Duplicates**: Write a function to sort an array that contains duplicate elements efficiently.
- **Sort an Array with Repeated Elements Using Counting Sort**: Write a function to sort an array that contains duplicate elements using the Counting Sort algorithm.

Cluster 1 - SQL Queries

```
-- 1. Select employees with salary greater than $50,000
SELECT * FROM employees WHERE salary > 50000;

-- 2. Select names and salaries of employees who earn more than the average salary
SELECT name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

-- 3. Select names and hire dates of employees hired in 2020
SELECT name, hire_date FROM employees WHERE YEAR(hire_date) = 2020;

-- 4. Select names and departments of employees in the 'Sales' department
SELECT name, department FROM employees WHERE department = 'Sales';

-- 5. Select names and salaries of employees earning more than $60,000 in 'Engineering'
SELECT name, salary FROM employees WHERE salary > 60000 AND department = 'Engineering';

-- 6. Select names and hire dates of employees hired before January 1, 2020
SELECT name, hire_date FROM employees WHERE hire_date < '2020-01-01';
```

Cluster 1 - SQL Queries

```
-- 7. Select names and salaries of employees earning more than $50,000 with 5+ years in the company
SELECT name, salary FROM employees WHERE salary > 50000 AND DATEDIFF(CURDATE(), hire_date) /
365 > 5;

-- 8. Select names and departments of employees in 'Marketing' earning more than $70,000
SELECT name, department FROM employees WHERE department = 'Marketing' AND salary > 70000;

-- 9. Select names and salaries of employees earning between $50,000 and $70,000
SELECT name, salary FROM employees WHERE salary BETWEEN 50000 AND 70000;

-- 10. Select names and hire dates of employees hired in January
SELECT name, hire_date FROM employees WHERE MONTH(hire_date) = 1;

-- 11. Select names and departments of employees in 'Finance' with 3+ years in the company
SELECT name, department FROM employees WHERE department = 'Finance' AND DATEDIFF(CURDATE(),
hire_date) / 365 > 3;
```

Cluster 1 - SQL Queries

```sql
-- 12. Select names and salaries of employees earning more than $60,000 with 4+ years in the company
SELECT name, salary FROM employees WHERE salary > 60000 AND DATEDIFF(CURDATE(), hire_date) /
365 > 4;

-- 13. Select names and departments of employees in 'HR' earning more than $55,000
SELECT name, department FROM employees WHERE department = 'HR' AND salary > 55000;

-- 14. Select names and hire dates of employees hired in 2019
SELECT name, hire_date FROM employees WHERE YEAR(hire_date) = 2019;

-- 15. Select names and salaries of employees earning more than $50,000 with 6+ years in the company
SELECT name, salary FROM employees WHERE salary > 50000 AND DATEDIFF(CURDATE(), hire_date) /
365 > 6;

-- 16. Select names and departments of employees in 'IT' earning more than $65,000
SELECT name, department FROM employees WHERE department = 'IT' AND salary > 65000;
```

Cluster 1 - SQL Queries

```
-- 17. Select names and salaries of employees earning between $45,000 and $65,000
SELECT name, salary FROM employees WHERE salary BETWEEN 45000 AND 65000;

-- 18. Select names and hire dates of employees hired in December
SELECT name, hire_date FROM employees WHERE MONTH(hire_date) = 12;

-- 19. Select names and departments of employees in 'Operations' with 2+ years in the company
SELECT name, department FROM employees WHERE department = 'Operations' AND
DATEDIFF(CURDATE(), hire_date) / 365 > 2;

-- 20. Select names and salaries of employees earning more than $55,000 with 7+ years in the company
SELECT name, salary FROM employees WHERE salary > 55000 AND DATEDIFF(CURDATE(), hire_date) /
365 > 7;

-- 21. Find the second highest salary
SELECT MAX(salary) AS second_highest_salary FROM employees WHERE salary < (SELECT MAX(salary)
FROM employees);
```

Cluster 1 - SQL Queries

```
-- 22. List employees earning more than the average salary
SELECT name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

-- 23. Find the total number of employees in each department
SELECT department, COUNT(*) AS total_employees FROM employees GROUP BY department;

-- 24. Get names of employees who have the same manager
SELECT name FROM employees WHERE manager_id = (SELECT manager_id FROM employees WHERE id = 100);

-- 25. List all employees who do not have a manager
SELECT name FROM employees WHERE manager_id IS NULL;

-- 26. Find names of employees who are not in the 'Sales' department
SELECT name FROM employees WHERE department != 'Sales';
```

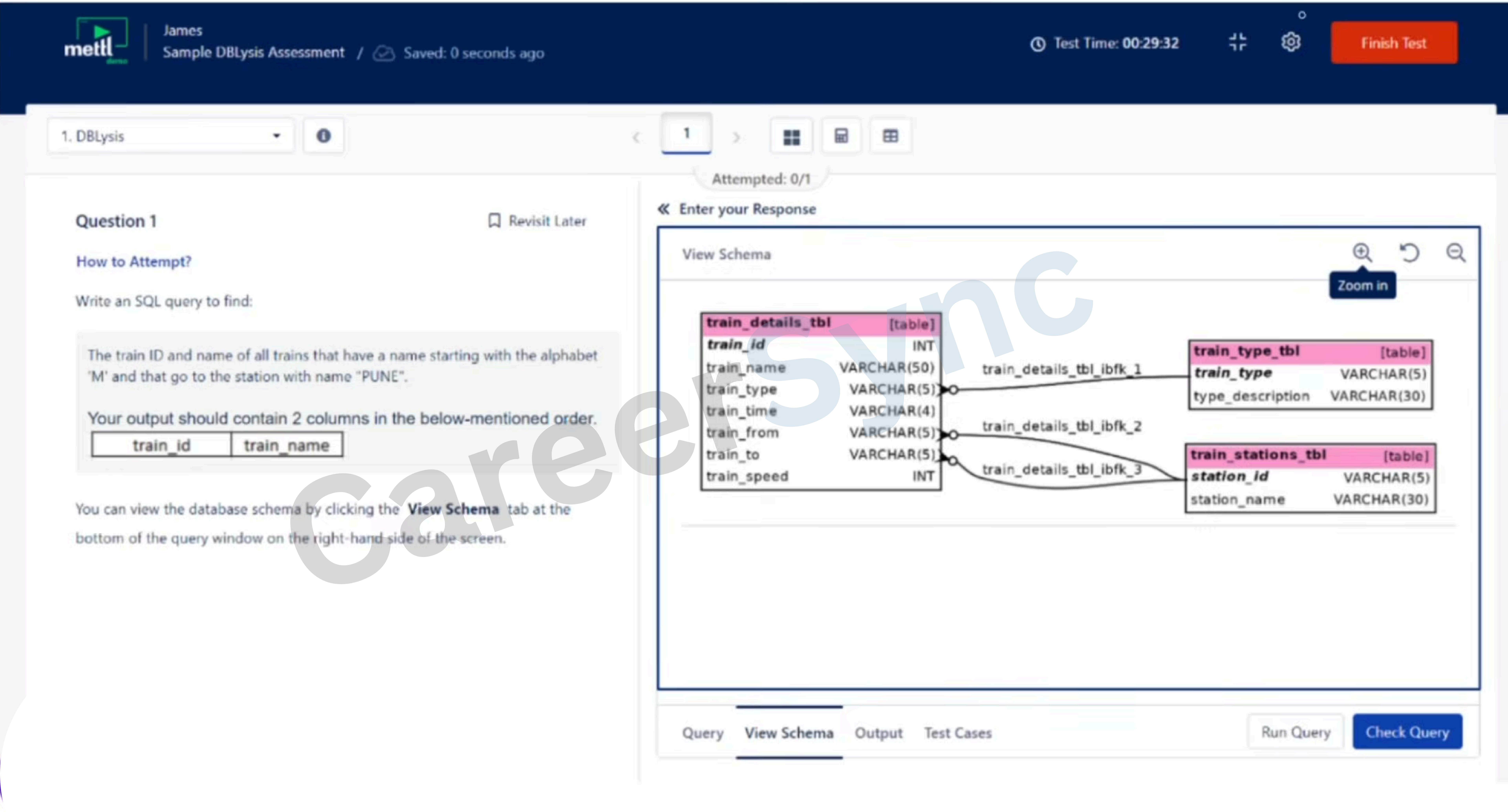
Cluster 1 - SQL Queries

```
-- 27. Find names of employees hired in the last 30 days
SELECT name FROM employees WHERE hire_date >= DATE_SUB(CURDATE(), INTERVAL 30 DAY);

-- 28. Find names of employees earning more than the average salary of their department
SELECT name FROM employees e WHERE salary > (SELECT AVG(salary) FROM employees WHERE
department = e.department);

-- 29. Find names of employees hired before a specific date
SELECT name FROM employees WHERE hire_date < '2020-01-01';

-- 30. Find names of employees with the same job title as employee with ID 100
SELECT name FROM employees WHERE job_title = (SELECT job_title FROM employees WHERE id = 100);
```



Question 1

How to Attempt?

Write an SQL query to find:

The train ID and name of all trains that have a name starting with the alphabet 'M' and that go to the station with name "PUNE".

Your output should contain 2 columns in the below-mentioned order.

train_id	train_name

You can view the database schema by clicking the **View Schema** tab at the bottom of the query window on the right-hand side of the screen.

train_details_tbl	[table]
train_id	INT
train_name	VARCHAR(50)
train_type	VARCHAR(5)
train_time	VARCHAR(4)
train_from	VARCHAR(5)
train_to	VARCHAR(5)
train_speed	INT

train_type_tbl	[table]
train_type	VARCHAR(5)
type_description	VARCHAR(30)

train_stations_tbl	[table]
station_id	VARCHAR(5)
station_name	VARCHAR(30)

Cluster 1 - Web Based Questions

1. Write a JavaScript function to toggle the visibility of an element on a webpage when a button is clicked.
2. Create a CSS animation that changes the background color of an element over a period of 5 seconds.
3. Write a JavaScript function to validate an email address entered in an input field.
4. Use CSS to create a responsive navigation bar that collapses into a hamburger menu on smaller screens.
5. Write a JavaScript function to dynamically add a new item to an unordered list on a webpage.
6. JavaScript Events: Describe the difference between onclick and addEventListener in JavaScript.
7. CSS Grid: How would you create a 3x3 grid layout using CSS Grid?
8. JavaScript DOM Manipulation: How can you change the text content of an HTML element using JavaScript?
9. CSS Media Queries: Write a media query to change the font size of a paragraph to 16px when the screen width is less than 600px.

Cluster 2

Technical Assessment Preparation:

- Focus on Python, ANSI SQL, and Cloud Fundamentals.
- The coding round will include Python questions of easy to medium difficulty.
- Practice writing SQL queries to strengthen your database skills.

Preparation Tips:

- Develop a strong understanding of Python programming concepts.
- Gain knowledge in Cloud Computing Fundamentals.
- Regularly practice coding problems and SQL queries to improve your problem-solving and technical skills.

Cluster 2 Coding Questions

- Python Program to Reverse a String: Write a Python program to reverse a given string.
- Python Program to Check Prime Number: Write a Python program to check if a given number is prime.
- Python Program to Find Factorial: Write a Python program to find the factorial of a given number.
- Python Program to Generate Fibonacci Series: Write a Python program to generate the Fibonacci series up to a given number.
- Python Program to Check Palindrome: Write a Python program to check if a given string is a palindrome.
- Python Program to Find Second Largest Number in Array: Write a Python program to find the second largest number in an array.
- Python Program to Sort Dictionary by Value: Write a Python program to sort a dictionary by its values.
- Python Program to Count Vowels in String: Write a Python program to count the number of vowels in a given string.
- Python Program to Find Common Elements in Two Lists: Write a Python program to find common elements in two lists.

Cluster 2 Coding Questions

- Python Program to Check Leap Year: Write a Python program to check if a given year is a leap year.
- Python Program to Check Armstrong Number: Write a Python program to check if a given number is an Armstrong number.
- Python Program to Find GCD of Two Numbers: Write a Python program to find the greatest common divisor (GCD) of two numbers.
- Python Program to Find LCM of Two Numbers: Write a Python program to find the least common multiple (LCM) of two numbers.
- Python Program to Find Sum of Digits of a Number: Write a Python program to find the sum of digits of a given number.
- Python Program to Check if a Number is Even or Odd: Write a Python program to check if a given number is even or odd.
- Python Program to Convert Temperature from Celsius to Fahrenheit: Write a Python program to convert temperature from Celsius to Fahrenheit.
- Python Program to Find Maximum and Minimum in List: Write a Python program to find the maximum and minimum elements in a list.

Cluster 2 Coding Questions

- Python Program to Count Occurrences of a Character in String: Write a Python program to count the occurrences of a character in a given string.
- Python Program to Remove Duplicates from List: Write a Python program to remove duplicates from a list.
- Python Program to Check if a String is a Substring of Another String: Write a Python program to check if one string is a substring of another string.
- Python Program to Find Length of a String Without Using len() Function: Write a Python program to find the length of a string without using the len() function.
- Python Program to Reverse a List: Write a Python program to reverse a list.
- Python Program to Find Sum of Geometric Series: Write a Python program to find the sum of a geometric series.
- Python Program to Find N-th Term of Geometric Progression Series: Write a Python program to find the N-th term of a geometric progression series.
- Python Program to Find Area of a Circle: Write a Python program to find the area of a circle.

Cluster 2 - SQL Queries

```
-- 1. Employee with the highest salary in each department
SELECT department, MAX(salary) AS highest_salary FROM employees GROUP BY department;

-- 2. Employees with the same salary
SELECT e1.name, e1.salary
FROM employees e1
JOIN employees e2
ON e1.salary = e2.salary AND e1.id != e2.id;

-- 3. Employees earning higher than the average salary of their department
SELECT name, salary FROM employees e
WHERE salary > (SELECT AVG(salary) FROM employees WHERE department = e.department);

-- 4. Employees earning lower than the average salary of their department
SELECT name, salary FROM employees e
WHERE salary < (SELECT AVG(salary) FROM employees WHERE department = e.department);
```

Cluster 2 - SQL Queries

```
-- 5. Employees earning equal to the average salary of their department
SELECT name, salary FROM employees e
WHERE salary = (SELECT AVG(salary) FROM employees WHERE department = e.department);

-- 6. Employees earning higher than the average salary of the company
SELECT name, salary FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);

-- 7. Employees earning lower than the average salary of the company
SELECT name, salary FROM employees
WHERE salary < (SELECT AVG(salary) FROM employees);

-- 8. Employees earning equal to the average salary of the company
SELECT name, salary FROM employees
WHERE salary = (SELECT AVG(salary) FROM employees);
```

Cluster 2 - SQL Queries

```
-- 9. Employees earning higher than the median salary of their department
SELECT name, salary FROM employees e
WHERE salary > (
   SELECT salary FROM employees
   WHERE department = e.department
   ORDER BY salary
   LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees WHERE department = e.department) / 2
);

-- 10. Employees earning lower than the median salary of their department
SELECT name, salary FROM employees e
WHERE salary < (
   SELECT salary FROM employees
   WHERE department = e.department
   ORDER BY salary
   LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees WHERE department = e.department) / 2
);
```

Cluster 2 - SQL Queries

```
-- 11. Employees earning equal to the median salary of their department
SELECT name, salary FROM employees e
WHERE salary = (
   SELECT salary FROM employees
   WHERE department = e.department
   ORDER BY salary
   LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees WHERE department = e.department) / 2
);

-- 12. Employees earning higher than the median salary of the company
SELECT name, salary FROM employees
WHERE salary > (
   SELECT salary FROM employees
   ORDER BY salary
   LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees) / 2
);
```

Cluster 2 - SQL Queries

```
-- 13. Employees earning lower than the median salary of the company
SELECT name, salary FROM employees
WHERE salary < (
  SELECT salary FROM employees
  ORDER BY salary
  LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees) / 2
);

-- 14. Employees earning equal to the median salary of the company
SELECT name, salary FROM employees
WHERE salary = (
  SELECT salary FROM employees
  ORDER BY salary
  LIMIT 1 OFFSET (SELECT COUNT(*) FROM employees) / 2
);
```

Cluster 2 - SQL Queries

```
-- 15. Employees earning higher than the mode salary of their department
SELECT name, salary FROM employees e
WHERE salary > (
   SELECT salary FROM employees
   WHERE department = e.department
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);

-- 16. Employees earning lower than the mode salary of their department
SELECT name, salary FROM employees e
WHERE salary < (
   SELECT salary FROM employees
   WHERE department = e.department
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);
```

Cluster 2 - SQL Queries

```
-- 17. Employees earning equal to the mode salary of their department
SELECT name, salary FROM employees e
WHERE salary = (
   SELECT salary FROM employees
   WHERE department = e.department
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);

-- 18. Employees earning higher than the mode salary of the company
SELECT name, salary FROM employees
WHERE salary > (
   SELECT salary FROM employees
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);
```

Cluster 2 - SQL Queries

```sql
-- 19. Employees earning lower than the mode salary of the company
SELECT name, salary FROM employees
WHERE salary < (
   SELECT salary FROM employees
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);

-- 20. Employees earning equal to the mode salary of the company
SELECT name, salary FROM employees
WHERE salary = (
   SELECT salary FROM employees
   GROUP BY salary
   ORDER BY COUNT(*) DESC
   LIMIT 1
);
```

Cluster 2 - SQL Queries

```sql
-- 21. Employees working for more than 5 years
SELECT name, hire_date FROM employees
WHERE DATEDIFF(CURDATE(), hire_date) / 365 > 5;

-- 22. Employees working for less than 5 years
SELECT name, hire_date FROM employees
WHERE DATEDIFF(CURDATE(), hire_date) / 365 < 5;

-- 23. Employees working for exactly 5 years
SELECT name, hire_date FROM employees
WHERE DATEDIFF(CURDATE(), hire_date) / 365 = 5;

-- 24. Employees working for more than 10 years
SELECT name, hire_date FROM employees
WHERE DATEDIFF(CURDATE(), hire_date) / 365 > 10;

-- 25. Employees working for less than 10 years
SELECT name, hire_date FROM employees
WHERE DATEDIFF(CURDATE(), hire_date) / 365 < 10;
```

cognizant®

Cluster 3

Technical Assessment Preparation:

- Focus on C#, ANSI SQL, HTML, CSS, and JavaScript.
- Emphasis on C# and ANSI SQL as core topics.
- The assessment includes coding questions with a 120-minute time limit.

Preparation Tips:

- Improve your debugging skills to identify and fix errors.
- Practice identifying and resolving logical errors and syntax errors in code.
- Strengthen your knowledge of SQL queries and basic web development concepts.

Cluster 3 Coding Questions

- Write a program to calculate the fuel consumption of a truck. The program should ask the user to enter the quantity of diesel to fill up the tank and the distance covered until the tank goes dry. Calculate the fuel consumption and display it in the format (liters per 100 kilometers). Convert the same result to the U.S. style of miles per gallon and display the result. If the quantity or distance is zero or negative, display "is an Invalid Input".
- Write a program to find the third largest element from an array using sorting techniques.
- Write a class FindFactor.java to find the factors of a given number. If the input is negative, ignore the sign and provide the output. If the input is zero, the output should be "No Factors".
- Write a program to calculate the increment in salary based on the appraisal rating. If the appraisal rating is between 1 and 3, the increment is 10% of the salary. If the appraisal rating is between 3.1 and 4, the increment is 25% of the salary. If the appraisal rating is between 4.1 and 5, the increment is 30% of the salary. If either the salary is 0 or negative or if the appraisal rating is not in the range 1 to 5 (inclusive), then the output should be "Invalid Input".

Cluster 3 Coding Questions

- Write a program to determine the season based on the month entered. If the month is December, January, or February, the season is Winter. If the month is March, April, or May, the season is Spring. If the month is June, July, or August, the season is Summer. If the month is September, October, or November, the season is Winter. If the month is invalid, print "Invalid month".
- Write a Java program to print all prime numbers in the interval [a, b] (a and b, both inclusive). If the input conditions are not met, display "Provide valid input".
- Write a program to find the maximum mark in each semester. If the entered mark is invalid (not between 0 and 100), display "You have entered invalid marks".
- Write a program to find the second largest element in an array.
- Write a program to find the sum of elements in a given array.
- Write a program to find the element with the left side smaller and the right side greater.
- Write a program to find the missing number in an array.
- Write a program to count the number of elements in two arrays.
- Write a program to find elements in a given range.
- Write a program to find the last duplicate element in a sorted array.

Cluster 3 Coding Questions

- Write a program to find the average of elements in an array.
- Write a program to find all symmetric pairs in an array of pairs.
- Write a program to cyclically rotate an array by one.
- Write a program to find the LCM of two numbers.
- Write a program to check if a number is an Armstrong number.
- Write a program to generate Fibonacci numbers.
- Write a program to check whether a given number is even or odd.
- Write a program to check if a given number is prime.
- Write a program to check if a given year is a leap year.
- Write a program to find the factorial of a number.
- Write a program to find the sum of a geometric series.
- Write a program to find the N-th term of a geometric progression series.
- Write a program to find the area of a circle.

Cluster 3 - SQL Queries

```
-- 1. Select all columns from the employees table
SELECT * FROM employees;

-- 2. Select the name and salary columns from the employees table
SELECT name, salary FROM employees;

-- 3. Select all employees whose salary is greater than 50000
SELECT * FROM employees WHERE salary > 50000;

-- 4. Select all employees whose name starts with 'A'
SELECT * FROM employees WHERE name LIKE 'A%';

-- 5. Select all employees whose name contains the substring 'an'
SELECT * FROM employees WHERE name LIKE '%an%';

-- 6. Select all employees whose name ends with 'n'
SELECT * FROM employees WHERE name LIKE '%n';
```

Cluster 3 - SQL Queries

```
-- 7. Select the top 5 employees with the highest salaries
SELECT * FROM employees ORDER BY salary DESC LIMIT 5;

-- 8. Select the average salary of all employees
SELECT AVG(salary) AS average_salary FROM employees;

-- 9. Select the maximum salary from the employees table
SELECT MAX(salary) AS max_salary FROM employees;

-- 10. Select the minimum salary from the employees table
SELECT MIN(salary) AS min_salary FROM employees;

-- 11. Select the total number of employees
SELECT COUNT(*) AS total_employees FROM employees;

-- 12. Select the number of employees in each department
SELECT department, COUNT(*) AS employee_count FROM employees GROUP BY department;
```

Cluster 3 - SQL Queries

```
-- 13. Select the number of employees who joined in the year 2020
SELECT COUNT(*) AS joined_in_2020 FROM employees WHERE YEAR(hire_date) = 2020;

-- 14. Select the names of employees who have a manager with ID 101
SELECT name FROM employees WHERE manager_id = 101;

-- 15. Select the names of employees who do not have a manager
SELECT name FROM employees WHERE manager_id IS NULL;

-- 16. Select the names of employees who have a salary between 50000 and 60000
SELECT name FROM employees WHERE salary BETWEEN 50000 AND 60000;

-- 17. Select the names of employees who have a salary greater than the average salary
SELECT name FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

-- 18. Select the names of employees who have a salary less than the average salary
SELECT name FROM employees WHERE salary < (SELECT AVG(salary) FROM employees);
```

Cluster 3 - SQL Queries

```sql
-- 19. Select the names of employees who have a salary equal to the average salary
SELECT name FROM employees WHERE salary = (SELECT AVG(salary) FROM employees);

-- 20. Select the names of employees who have a salary greater than the salary of employee with ID 102
SELECT name FROM employees WHERE salary > (SELECT salary FROM employees WHERE id = 102);

-- 21. Select the names of employees who have a salary less than the salary of employee with ID 102
SELECT name FROM employees WHERE salary < (SELECT salary FROM employees WHERE id = 102);

-- 22. Select the names of employees who have a salary equal to the salary of employee with ID 102
SELECT name FROM employees WHERE salary = (SELECT salary FROM employees WHERE id = 102);

-- 23. Select the names of employees who have a salary greater than the maximum salary of employees in department 10
SELECT name FROM employees WHERE salary > (SELECT MAX(salary) FROM employees WHERE department = 10);
```

Cluster 3 - SQL Queries

```sql
-- 24. Select the names of employees who have a salary less than the minimum salary of employees in department 10
SELECT name FROM employees WHERE salary < (SELECT MIN(salary) FROM employees WHERE department = 10);
```

```sql
-- 25. Select the names of employees who have a salary equal to the maximum salary of employees in department 10
SELECT name FROM employees WHERE salary = (SELECT MAX(salary) FROM employees WHERE department = 10);
```

Cluster 3 - Web Based Questions

- **HTML Structure**: Create an HTML document that includes a header, a navigation bar, a main content section, and a footer. Ensure the navigation bar has links to different sections of the page.
- **CSS Styling**: Write CSS to style a button that changes color when hovered over and has a shadow effect when clicked.
- **JavaScript Functionality**: Implement a JavaScript function that toggles the visibility of a hidden div when a button is clicked.
- **Responsive Design**: Use CSS media queries to adjust the layout of a webpage for different screen sizes, ensuring that the navigation bar changes from horizontal to vertical on smaller screens.
- **DOM Manipulation**: Write a JavaScript function that dynamically adds a new list item to an unordered list when a button is clicked.

1.Write a program to calculate the fuel consumption of your truck.The program should ask the user to enter the quantity of diesel to fill up the tank and the distance covered till the tank goes dry.Calculate the fuel consumption and display it in the format (liters per 100 kilometers).
Convert the same result to the U.S. style of miles per gallon and display the result. If the quantity or distance is zero or negative display ” is an Invalid Input”.
[Note: The US approach of fuel consumption calculation (distance / fuel) is the inverse of the European approach (fuel / distance). Also note that 1 kilometer is 0.6214 miles, and 1 liter is 0.2642 gallons.]
The result should be with two decimal place.To get two decimal place refer the below-mentioned print statement :
float cost=670.23;
System.out.printf(“You need a sum of Rs.%.2f to cover the trip”,cost);

Sample Input 1:

- Enter the no of liters to fill the tank
  20
- Enter the distance covered
  150

Sample Output 1:

- Liters/100KM
  13.33
- Miles/gallons
  17.64

Explanation:

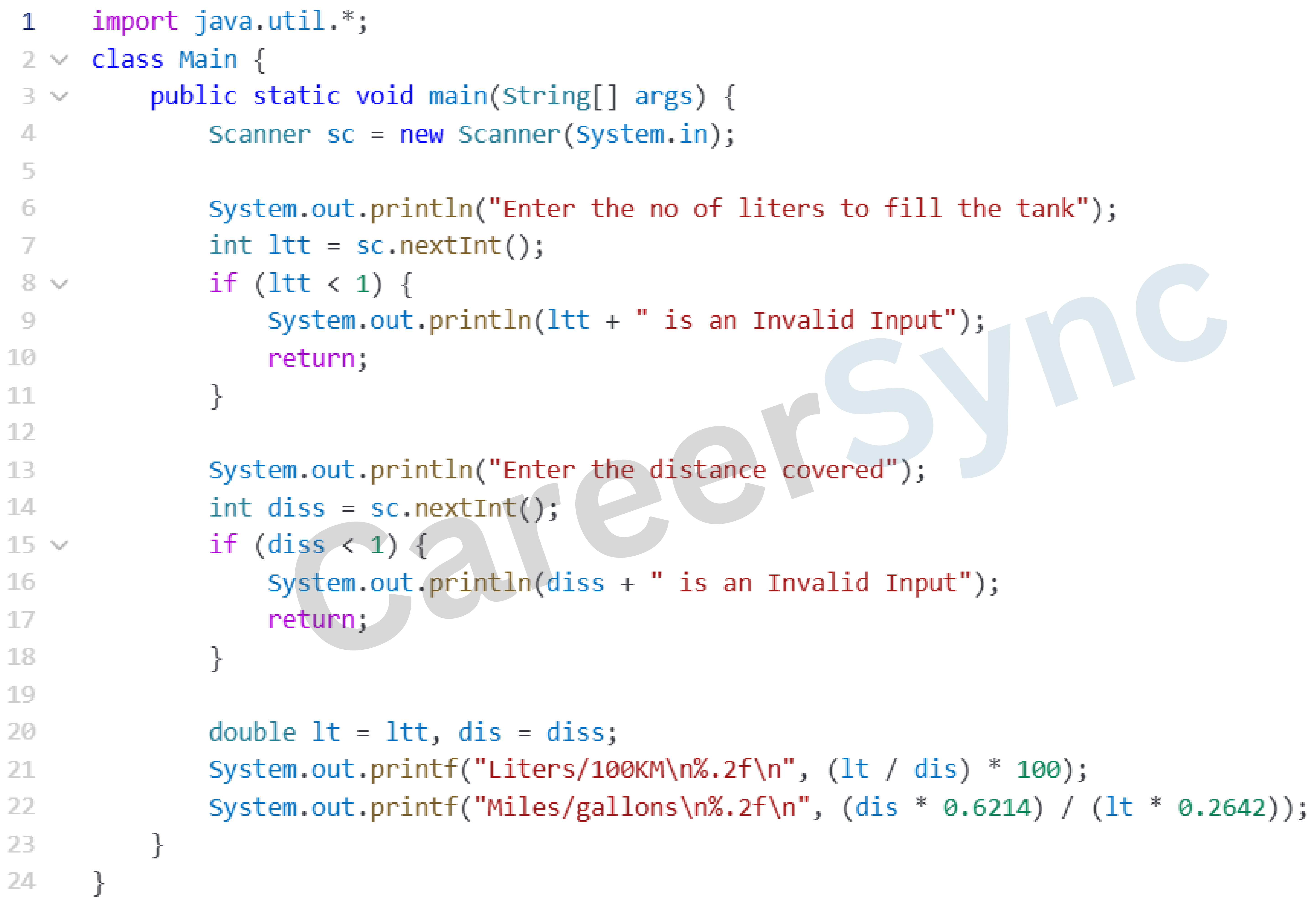
- For 150 KM fuel consumption is 20 liters,
- Then for 100 KM fuel consumption would be (20/150)*100=13.33,
- Distance is given in KM, we have to convert it to miles (150*0.6214)=93.21,
- Fuel consumption is given in liters, we have to convert it to gallons (20*0.2642)=5.284,
- Then find (miles/gallons)=(93.21/5.284)=17.64

Sample Input 2:

- Enter the no of liters to fill the tank
  -5

Sample Output 2:

- -5 is an Invalid Input



```java
import java.util.*;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the no of liters to fill the tank");
        int ltt = sc.nextInt();
        if (ltt < 1) {
            System.out.println(ltt + " is an Invalid Input");
            return;
        }

        System.out.println("Enter the distance covered");
        int diss = sc.nextInt();
        if (diss < 1) {
            System.out.println(diss + " is an Invalid Input");
            return;
        }

        double lt = ltt, dis = diss;
        System.out.printf("Liters/100KM\n%.2f\n", (lt / dis) * 100);
        System.out.printf("Miles/gallons\n%.2f\n", (dis * 0.6214) / (lt * 0.2642));
    }
}
```

2.Problem Statement – Vohra went to a movie with his friends in a Wave theatre and during break time he bought pizzas, puffs and cool drinks. Consider the following prices :

- **Rs.100/pizza**
- **Rs.20/puffs**
- **Rs.10/cooldrink**

Generate a bill for What Vohra has bought.

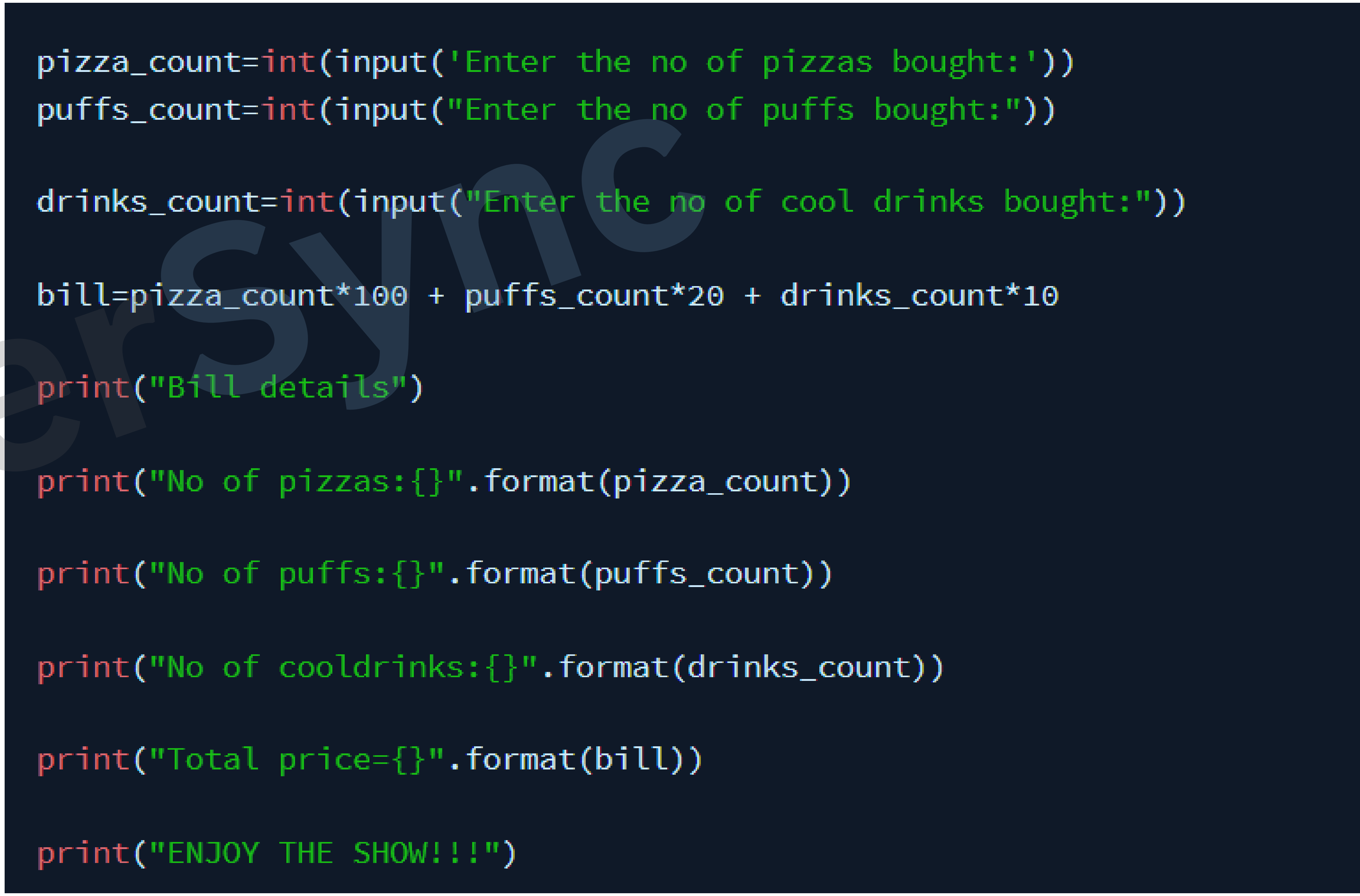
Sample Input 1:

- Enter the no of pizzas bought:10
- Enter the no of puffs bought:12
- Enter the no of cool drinks bought:5

Sample Output 1:

Bill Details

- No of pizzas:10
- No of puffs:12
- No of cooldrinks:5
- Total price=1290



```
pizza_count=int(input('Enter the no of pizzas bought:'))
puffs_count=int(input("Enter the no of puffs bought:"))

drinks_count=int(input("Enter the no of cool drinks bought:"))

bill=pizza_count*100 + puffs_count*20 + drinks_count*10

print("Bill details")

print("No of pizzas:{}".format(pizza_count))

print("No of puffs:{}".format(puffs_count))

print("No of cooldrinks:{}".format(drinks_count))

print("Total price={}".format(bill))

print("ENJOY THE SHOW!!!")
```

3.Problem Statement – Ritik wants a magic board, which displays a character for a corresponding number for his science project. Help him to develop such an application.

For example when the digits 65,66,67,68 are entered, the alphabet ABCD are to be displayed.

[Assume the number of inputs should be always 4]

Sample Input 1:

- Enter the digits:
- 65
- 66
- 67
- 68

Sample Output 1:

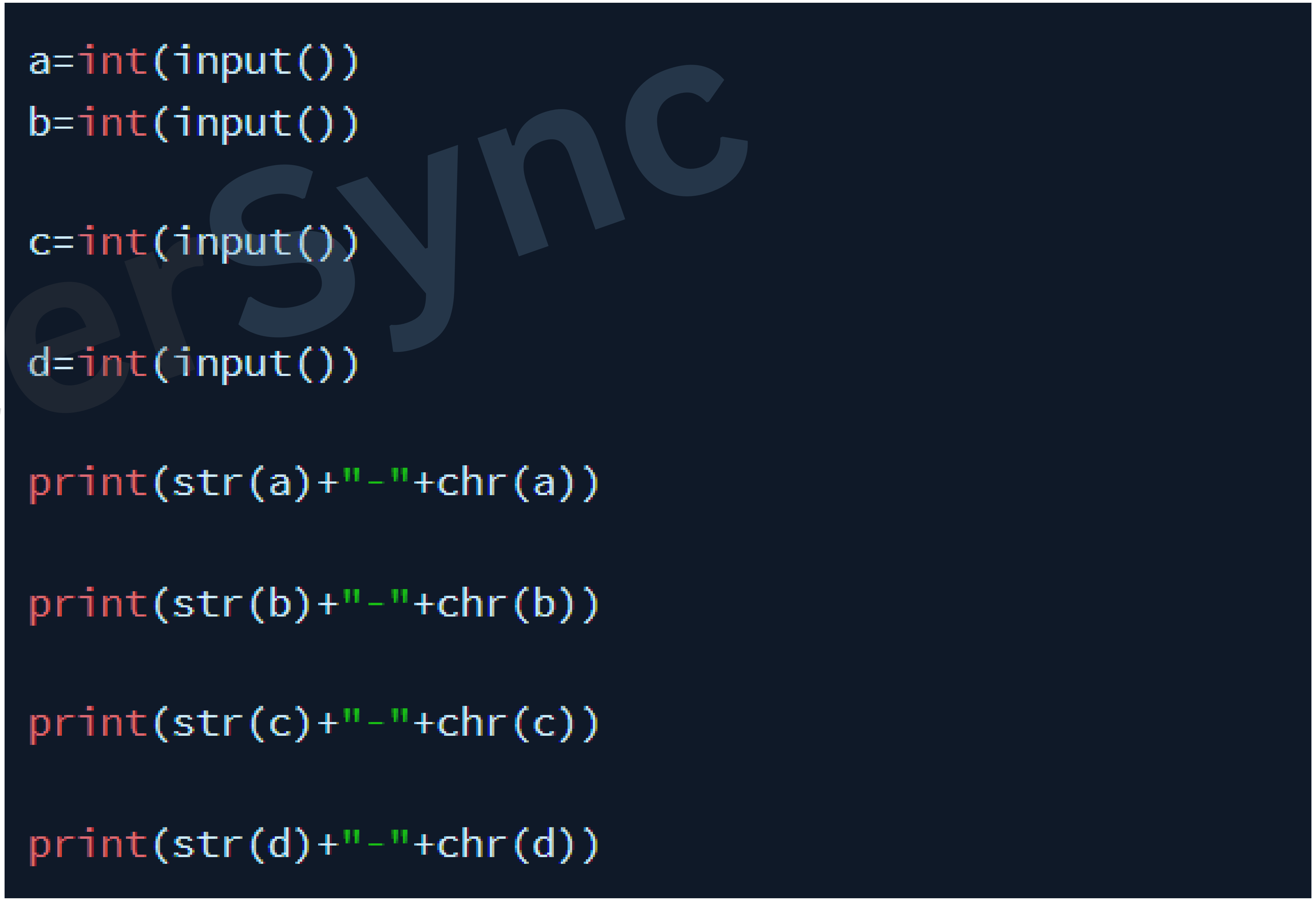
```
65-A
66-B
67-C
68-D
```

Sample Input 2:

- Enter the digits:
- 115
- 116
- 101
- 112

Sample Output 2:

```
115-s
116-t
101-e
112-p
```



```python
a=int(input())
b=int(input())

c=int(input())

d=int(input())

print(str(a)+"-"+chr(a))

print(str(b)+"-"+chr(b))

print(str(c)+"-"+chr(c))

print(str(d)+"-"+chr(d))
```

CareerSync

tusind tak

ngiyabonga

dziękuję

谢谢 dakujem vám

ありがとう

merci

suksema

danke

thank

you

baie dankie

धन्यवाद molte grazie

gracias

takk

obrigada

obrigado

dank u

teşekkür edire

mahalo

gràcies

teşekkür ederim

شكرا

tänan

tack så mycket

CareerSync