Task-1 SQL: To generate the desired report using SQL, you can execute the following query:

|  |
| --- |
| SQL |
| SELECT DEPT\_NAME, AVG(MONTHLY\_SALARY) AS AVG\_MONTHLY\_SALARY  FROM ASDE\_Assignment\_Table  GROUP BY DEPT\_NAME  ORDER BY AVG\_MONTHLY\_SALARY DESC  LIMIT 3; |

Assuming you have created a table named "ASDE\_Assignment\_Table" that corresponds to the data in the attached file, this query will retrieve the top 3 departments along with their names and average monthly salaries.

Task-2 Scripting: To generate the same report using CSV files and a scripting language like Python, you can use the following script:

|  |
| --- |
| PYTHON |
| import csv  # Define the path to the CSV file  csv\_file\_path = "path/to/your/csv/file.csv"  # Read the CSV file  with open(csv\_file\_path, "r") as file:  reader = csv.DictReader(file)  data = [row for row in reader]  # Group the data by department and calculate the average monthly salary  department\_salaries = {}  for row in data:  department = row["DEPT\_NAME"]  salary = float(row["MONTHLY\_SALARY"])  if department in department\_salaries:  department\_salaries[department].append(salary)  else:  department\_salaries[department] = [salary]  # Calculate the average monthly salary for each department  average\_salaries = {}  for department, salaries in department\_salaries.items():  average\_salaries[department] = sum(salaries) / len(salaries)  # Sort the departments based on average monthly salary  sorted\_departments = sorted(average\_salaries.items(), key=lambda x: x[1], reverse=True)  # Fetch the top 3 departments  top\_3\_departments = sorted\_departments[:3]  # Print the report  for department, average\_salary in top\_3\_departments:  print(department)  print(f"{average\_salary:.2f} (USD)") |

Make sure to replace "path/to/your/csv/file.csv" with the actual path to your CSV file containing the department data.

Task-3 Debugging: The script provided below fixes the bugs in the given Bash/Python script:

|  |
| --- |
| BASH |
| #!/bin/bash  # Read the integer input  read -p "Enter an integer: " n  if (( n < 10 )); then  # Calculate the square of n  echo $(( n \* n ))  elif (( n >= 10 && n <= 20 )); then  # Calculate the factorial of (n-10)  fact=1  for (( i=1; i<=n-10; i++ )); do  fact=$(( fact \* i ))  done  echo $fact  else  # Calculate the sum of all integers between 1 and (n-20)  sum=0  for (( i=1; i<=n-20; i++ )); do  sum=$(( sum + i ))  done  echo $sum  fi |

zIn the original script, there were two issues:

1. The **for** loop in the factorial calculation started from 0 instead of 1. The fix is to change **for (( i=0; i<=n-10; i++ ))** to **for (( i=1; i<=n-10; i++ ))**.
2. The **for** loop in the sum calculation used **i<n-20** as the condition instead of **i<=n-20**. The fix is to change **for (( i=1; i<n-20; i++ ))** to `for (( i=1