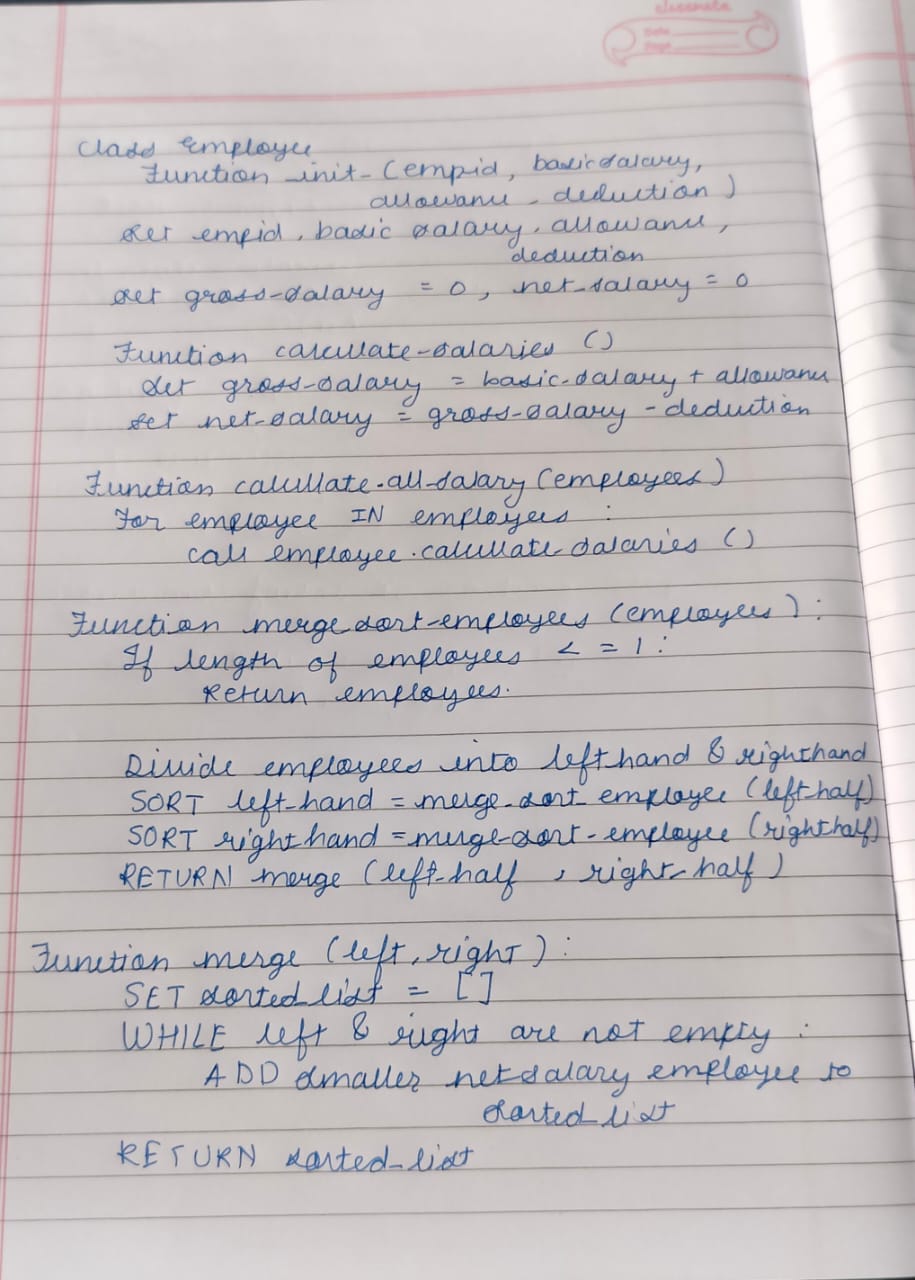
**EXPERIMENT NO. 3**

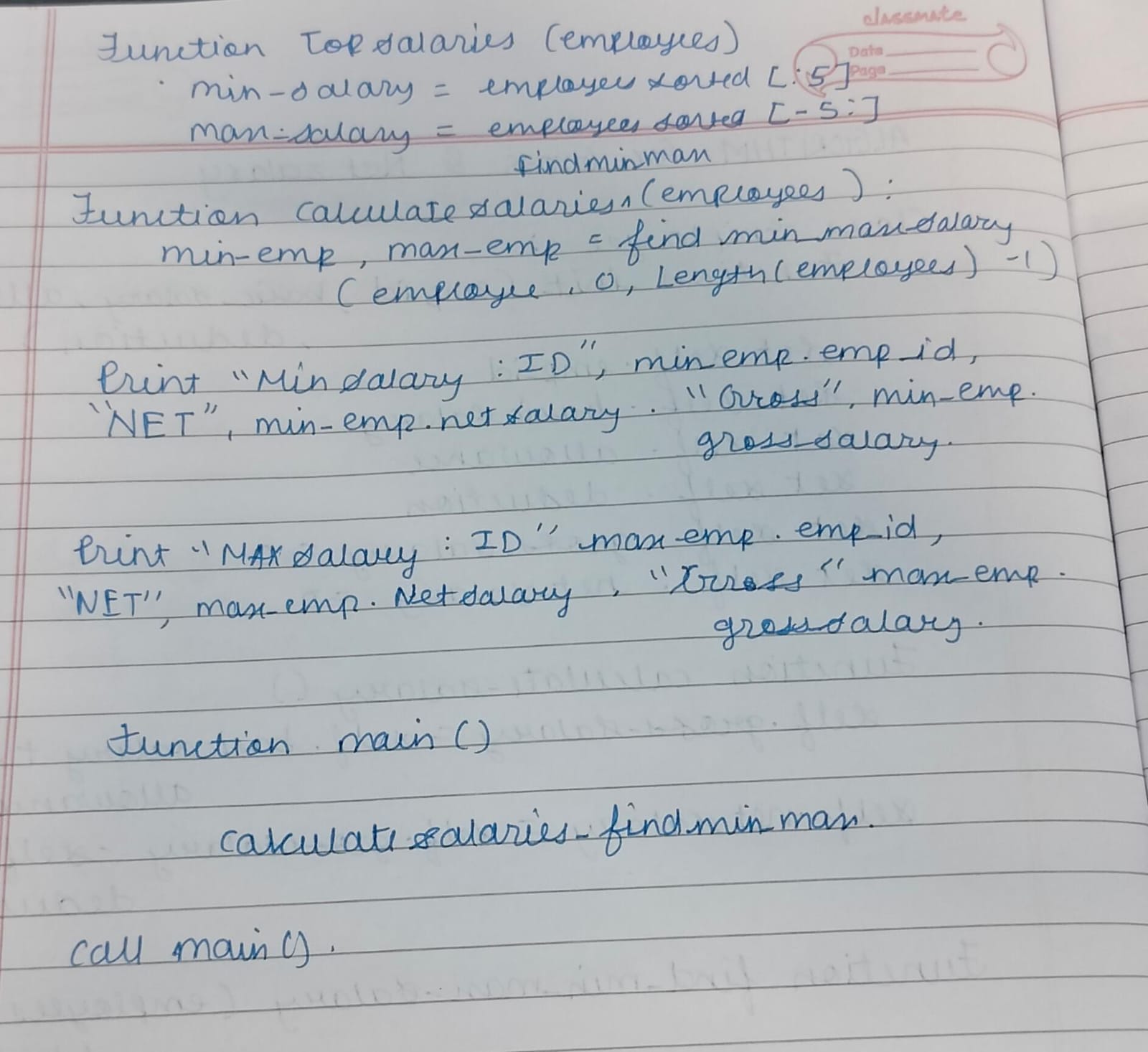
**NAME: ADITYA B. KULKARNI  
REGISTRATION NO : 241070908  
SUBJECT : DAA (LAB)  
SY BTECH COMPUTER ENGINEERING  
  
AIM :** Write an algorithm to find gross and net salary of employees.  
ABC co. ltd. has 2000 employees. calculate each employees salary and find employee with minimum salary and maximum salary.

**THEORY :**Divide and Conquer Algorithm is a problem-solving technique used to solve problems by dividing the main problem into subproblems, solving them individually and then merging them to find solution to the original problem

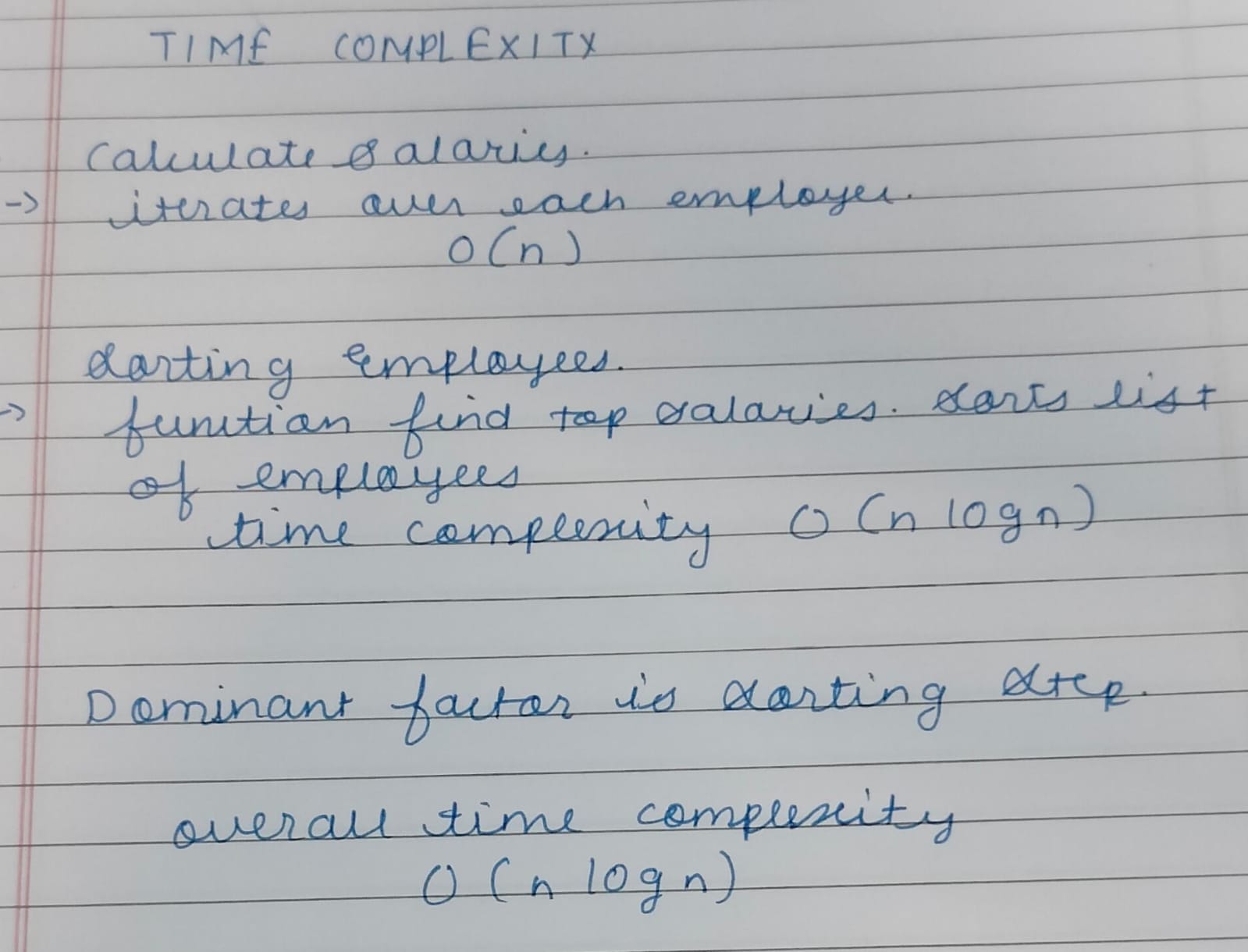
**Key Steps in the Divide and Conquer Approach**

1. **Divide:** Break the problem into smaller subproblems that are similar to the original problem but smaller in size.
2. **Conquer:** Solve the subproblems recursively. If the subproblems are small enough, solve them directly (base case).
3. **Combine:** Combine the solutions of the subproblems to form the solution to the original problem.

**Algorithm /** **Pseudocode :**



**TIME COMPLEXITY :**



**CODE :**

import csv

# Employee Class to represent basic details

class Employee:

    def \_\_init\_\_(self, emp\_id, basic\_salary, allowances, deductions):

        self.emp\_id = emp\_id

        self.basic\_salary = basic\_salary

        self.allowances = allowances

        self.deductions = deductions

        self.gross\_salary = 0

        self.net\_salary = 0

    # Method to calculate gross and net salary

    def calculate\_salaries(self):

        self.gross\_salary = self.basic\_salary + self.allowances

        self.net\_salary = self.gross\_salary - self.deductions

# Function to calculate salaries for all employees

def calculate\_all\_salaries(employees):

    for emp in employees:

        emp.calculate\_salaries()

# Divide and Conquer: Merge Sort Algorithm

def merge\_sort\_employees(employees):

    if len(employees) <= 1:

        return employees

    mid = len(employees) // 2

    # Divide the list into two halves

    left\_half = merge\_sort\_employees(employees[:mid])

    right\_half = merge\_sort\_employees(employees[mid:])

    # Conquer: Merge the two halves

    return merge\_employees(left\_half, right\_half)

# Function to merge two sorted lists of employees based on net salary

def merge\_employees(left, right):

    sorted\_list = []

    i = j = 0

    # Merging two sorted halves

    while i < len(left) and j < len(right):

        if left[i].net\_salary < right[j].net\_salary:

            sorted\_list.append(left[i])

            i += 1

        else:

            sorted\_list.append(right[j])

            j += 1

    # Append remaining items in both halves

    sorted\_list.extend(left[i:])

    sorted\_list.extend(right[j:])

    return sorted\_list

# Function to find top 5 min and max salaries using merge sort

def find\_top\_salaries(employees):

    # Use divide-and-conquer merge sort to sort employees by net salary

    employees\_sorted = merge\_sort\_employees(employees)

    # Get top 5 min and max salaries

    min\_salaries = employees\_sorted[:5]

    max\_salaries = employees\_sorted[-5:]

    return min\_salaries, max\_salaries

# Function to output the results

def print\_salaries(salaries, title):

    print(f"{title}:")

    for emp in salaries:

        print(f"ID {emp.emp\_id}, Gross Salary: {emp.gross\_salary}, Net Salary: {emp.net\_salary}")

    print()

# Reading the CSV file and creating employee objects

employees = []

csv\_file\_path = 'emp.csv'  # Ensure the CSV file is in the same directory

with open(csv\_file\_path, mode='r') as file:

    reader = csv.DictReader(file)

    for row in reader:

        emp\_id = int(row["Emp\_ID"])

        basic\_salary = int(row["Basic\_Salary"])

        allowances = int(row["Allowances"])

        deductions = int(row["Deductions"])

        employees.append(Employee(emp\_id, basic\_salary, allowances, deductions))

# Calculate salaries for all employees

calculate\_all\_salaries(employees)

# Find top 5 min and max salaries using divide and conquer approach (merge sort)

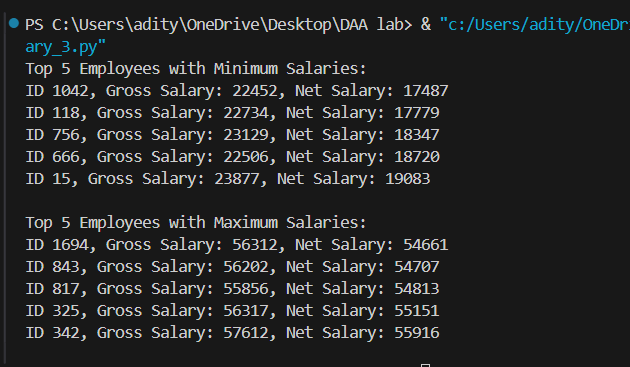
min\_salaries, max\_salaries = find\_top\_salaries(employees)

# Print results

print\_salaries(min\_salaries, "Top 5 Employees with Minimum Salaries")

print\_salaries(max\_salaries, "Top 5 Employees with Maximum Salaries")

**OUTPUT (POSITIVE AND NEGATIVE TEST CASES ) :**

****

**CONCLUSION :** Hence we have calculated the Gross and Net salary of the employees using  Divide and Conquer algorithm.