Code:

```python
"""
Author: Ayush Bothra
Date: 20-08-2024
Aim: Code for finding the maximum element in an unsorted array
using both iterative and divide and conquer approach
and get a solution in O(n) time.
"""

# Importing the necessary libraries
import pandas as pd
import numpy as np


class LoadSalaryData():
    def __init__(self, dataFrame):
    # extract data from the dataframe taken:
        self.df = pd.read_csv(dataFrame)
        self.basic_salary = self.df[self.df.columns[1]]
        self.house_rent = self.basic_salary * 0.5
        self.provident_fund = self.basic_salary * 0.12


class SalaryCalculator:
    def __init__(self, data):
        self.basic_salary = data.basic_salary
        self.house_rent = data.house_rent
        self.provident_fund = data.provident_fund

    def find_gross(self, index):
        self.gross_salary = self.basic_salary[index] + self.house_rent[index]
        return self.gross_salary

    def find_net(self, index):
        if self.gross_salary < 30000:
            self.income_tax = gross_salary * 0.05
        elif self.gross_salary < 41000:
            self.income_tax = self.gross_salary * 0.1
        else:
            self.income_tax = gross_salary * 0.15
        self.net_salary = self.gross_salary - self.income_tax -
self.provident_fund[index]
```

```
        return self.net_salary


class GetMinMax:
    def __init__(self, array):
        self.array = array

    def min_max_iterative(self):
        if len(self.array) == 0:
            return None, None, None, None
        if len(self.array) == 1:
            return 1, 1, self.array[0], self.array[0]
        int_min = self.array[0]
        int_max = self.array[0]
        low, high = 0, 0

        # Check for both maximum and minimum in a single iteration
        for i in range(len(self.array)):
            if self.array[i] <= int_min:
                int_min = self.array[i]
                low = i
            elif self.array[i] >= int_max:
                int_max = self.array[i]
                high = i

        return int_min, int_max, low, high

    def min_max_recursive(self, low, high):
        if len(self.array) == 0:
            return None, None, None, None
        # If there's only one element in the divided array
        if low == high:
            return self.array[low], self.array[high], low, high
        # If there are two elements in the divided array
        if high == low + 1:
            if self.array[high] > self.array[low]:
                return self.array[low], self.array[high], low, high
            else:
                return self.array[high], self.array[low], high, low

        # Compute the mid of the array
        mid = (low + high) // 2

        # Recursively traverse the right and left arrays while dividing them
```

```
        lmin, lmax, llow, lhigh = self.min_max_recursive(low, mid)
        rmin, rmax, rlow, rhigh = self.min_max_recursive(mid + 1, high)

        max_val = max(lmax, rmax)
        min_val = min(lmin, rmin)

        if max_val == rmax:
            high = rhigh
        else:
            high = lhigh

        if min_val == rmin:
            low = rlow
        else:
            low = llow

        return min_val, max_val, low, high


if __name__ == "__main__":
    df_storer = []

    for i in range(1, 6):
        df = (f'salaries_{i}.csv')
        df_storer.append(df)

    for df in df_storer:
        #  checking for negative values or empty array values first
        data = LoadSalaryData(df)
        if np.any(data.basic_salary < 0):
            print(f"Error: In {df}, negative values not allowed.")
            print('# ------------------------------------- #')
            continue
        elif np.any(np.isnan(data.basic_salary)):
            print(f"Error:In {df}, NaN values should not be present")
            print('# ------------------------------------- #')
            continue

        calculate_salary = SalaryCalculator(data)
        gross_salary_all = []
        net_salary_all = []

        for i in range(2000):
            gross_salary = calculate_salary.find_gross(i)
```

```
            gross_salary_all.append(gross_salary)
            net_salary = calculate_salary.find_net(i)
            net_salary_all.append(net_salary)

        min_max = GetMinMax(net_salary_all)

        print('Recursive answers:')

        min_recursive, max_recursive, recursive_low, recursive_high =
min_max.min_max_recursive(0, len(net_salary_all) - 1)
        print(f'''Minimum is: {min_recursive:.2f} at location
{recursive_low:.2f},
              Maximum is: {max_recursive:.2f} at location
{recursive_high:.2f}''')

        print('Iterative answers:')

        min_iterative, max_iterative, iterative_low, iterative_high =
min_max.min_max_iterative()
        print(f'''Minimum is: {min_iterative:.2f} at location
{iterative_low:.2f},
              Maximum is: {max_iterative:.2f} at location
{iterative_high:.2f}''')

        print('# ------------------------------------- #')
```

Output:

```
Recursive answers:
Minimum is: 24612.30 at location 411.00,
|   |   |   | Maximum is: 57745.38 at location 110.00
Iterative answers:
Minimum is: 24612.30 at location 411.00,
|   |   |   | Maximum is: 57745.38 at location 110.00
# ------------------------------------ #
Recursive answers:
Minimum is: 24619.68 at location 205.00,
|   |   |   | Maximum is: 57747.69 at location 1214.00
Iterative answers:
Minimum is: 24619.68 at location 205.00,
|   |   |   | Maximum is: 57747.69 at location 1214.00
# ------------------------------------ #
Recursive answers:
Minimum is: 24608.61 at location 1976.00,
|   |   |   | Maximum is: 57739.61 at location 291.00
Iterative answers:
Minimum is: 24608.61 at location 1976.00,
|   |   |   | Maximum is: 57739.61 at location 291.00
# ------------------------------------ #
Error: In salaries_4.csv, negative values not allowed.
# ------------------------------------ #
Error:In salaries_5.csv, NaN values should not be present
# ------------------------------------ #
```