

Name : Atharva Dhage

Roll no: 231070017

Batch : Computer A

## Lab - 3

### Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <ctime>
#include <vector>
using namespace std;

string generateRandomName() {
    vector<string> firstNames = {"John", "Jane", "Alex", "Emily", "Chris", "Katie", "Michael", "Sarah",
    "David", "Laura"};
    vector<string> lastNames = {"Smith", "Johnson", "Williams", "Brown", "Jones", "Miller", "Davis",
    "Garcia", "Wilson", "Martinez"};

    string firstName = firstNames[rand() % firstNames.size()];
    string lastName = lastNames[rand() % lastNames.size()];

    return firstName + " " + lastName;
}

int generateRandomSalary() {
    return rand() % 90001 + 10000; // Random salary between 10,000 and 100,000
}

int main() {
    srand(static_cast<unsigned int>(time(0))); // Seed for random number generation

    ofstream file("5input.csv");

    if (!file.is_open()) {
        cerr << "Error opening file!" << endl;
        return 1;
    }

    // Write the header
```

```

file << "Name,Salary\n";

// Generate and write 2000 records
for (int i = 0; i < 2000; ++i) {
    string name = generateRandomName();
    int salary = generateRandomSalary();
    file << name << "," << salary << "\n";
}

file.close();
cout << "CSV file created successfully!" << endl;

return 0;
}

```

```

#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <string>
using namespace std;

int findMax(vector<double> a, int lo, int hi)
{
    if (lo > hi)
        return -1;

    if (lo == hi)
        return a[lo];
    int mid = (lo + hi) / 2;
    int leftMax = findMax(a, lo, mid);
    int rightMax = findMax(a, mid + 1, hi);
    return max(leftMax, rightMax);
}

int findMin(vector<double> a, int lo, int hi)
{
    if (lo > hi)
        return -1 ;

    if (lo == hi)
        return a[lo];
    int mid = (lo + hi) / 2;
    int leftMin = findMin(a, lo, mid);
    int rightMin = findMin(a, mid + 1, hi);
    return min(leftMin, rightMin);
}

```

```

int main()
{
    ifstream inputFile("1input.csv"); // Input CSV file
    ofstream outputFile("1.csv"); // Output CSV file
    vector<double>v;
    if (!inputFile.is_open() || !outputFile.is_open())
    {
        cout << "Error opening file!" << endl;
    }

    string line;
    // Write the header for the output file
    outputFile << "Name,Salary,Tax,Home Rent,Bonus\n";

    // Skip the header line in the input file
    getline(inputFile, line);

    // Process each line
    while (getline(inputFile, line))
    {
        stringstream ss(line);
        string name, salaryStr;
        getline(ss, name, ',');
        getline(ss, salaryStr, ',');

        double salary = stod(salaryStr);
        v.push_back(salary);
        double tax = 0.10 * salary;
        double homeRent = 0.20 * salary;
        double bonus = 0.15 * salary;

        // Write the results to the output file
        outputFile << name << "," << salary << "," << tax << "," << homeRent
<< "," << bonus << "\n";
    }

    inputFile.close();
    outputFile.close();

    cout << "Calculations completed and output saved to
'output_finances.csv'." << endl;

    if findMin(v) < 0
        cout<<"Salary can't be negative";
}

```

```

    }
    else{
        cout<<"Min Salary -> "<<findMin(v,1,2000)<<endl;
        cout<<"Max Salary -> "<<findMax(v,1,2000);
    }

    return 0;
}

```

## Output

```

Calculations completed and output saved to 'output_finances.csv'.
Min Salary -> 10008
Max Salary -> 42755
PS C:\Users\HP\Desktop\c++>

Calculations completed and output saved to 'output_finances.csv'.
Min Salary -> 10019
Max Salary -> 42715
PS C:\Users\HP\Desktop\c++>

Error opening file!
Calculations completed and output saved to 'output_finances.csv'.
PS C:\Users\HP\Desktop\c++>

Calculations completed and output saved to 'output_finances.csv'.
Min Salary -> 10003
Max Salary -> 42702
PS C:\Users\HP\Desktop\c++>

Calculations completed and output saved to 'output_finances.csv'.
Salary can't be negative
PS C:\Users\HP\Desktop\c++>

```

## Conclusion

The Divide and Conquer algorithmic strategy is a powerful approach for solving complex problems by breaking them down into simpler subproblems, solving each independently, and then combining their solutions. We deal on a real world problem of handing files of a having a huge data of empoylee