**Experiment no. 5**

**Name: Sonali Dattatray Kaingade**

**PRN: 21620002**

**Title:** Find t and d weight of data.

**Code:**

#include <iostream>

#include <fstream>

#include <sstream>

#include <map>

using namespace std;

// A struct to represent a cell's data in the CSV file

struct CellData

{

    int count;

    int tWeight;

    int dWeight;

};

// Function to read data from the input CSV file into the provided data structures

void readData(const string &filename, map<string, map<string, CellData>> &cellData,

              map<string, int> &columnTotal, map<string, int> &rowTotal)

{

    fstream file(filename, ios::in);

    if (!file.is\_open())

    {

        cout << "Couldn't open file: " << filename << endl;

        return;

    }

    string line, row, col, count;

    int val;

    int lineNumber = 0;

    while (getline(file, line))

    {

        stringstream str(line);

        if (lineNumber == 0)

        {

            lineNumber++;

            continue; // Skip the header line

        }

        getline(str, row, ',');

        getline(str, col, ',');

        getline(str, count, ',');

        val = stoi(count);

        cellData[row][col].count += val;

        columnTotal[col] += val;

        rowTotal[row] += val;

    }

}

// Function to write the result to an output CSV file

void writeResult(const string &filename, const map<string, map<string, CellData>> &cellData,

                 const map<string, int> &columnTotal, const map<string, int> &rowTotal)

{

    ofstream fw(filename, ios::out);

    fw << "Column\\Row, Count, T-Weight, D-Weight, Count, T-Weight, D-Weight, Count, T-Weight, D-Weight" << endl;

    int total1 = 0;

    for (const auto &rowEntry : rowTotal)

    {

        total1 += rowEntry.second;

    }

    for (const auto &rowEntry : rowTotal)

    {

        const string &row = rowEntry.first;

        fw << row << ",";

        for (const auto &colEntry : columnTotal)

        {

            const string &col = colEntry.first;

            const CellData &cell = cellData.at(row).at(col);

            fw << cell.count << ",";

            fw << ((float)cell.count / rowTotal.at(row)) \* 100 << "%,";

            fw << ((float)cell.count / colEntry.second) \* 100 << "%,";

        }

        fw << rowTotal.at(row) << ","

           << "100%," << ((float)rowTotal.at(row) / total1) \* 100 << "%" << endl;

    }

    fw << "Total,";

    int total = 0;

    for (const auto &colEntry : columnTotal)

    {

        total += colEntry.second;

    }

    for (const auto &colEntry : columnTotal)

    {

        fw << colEntry.second << ",";

        fw << ((float)colEntry.second / total) \* 100 << "%,";

        fw << "100%,";

    }

    fw << total << ",100%, 100%" << endl;

}

int main()

{

    map<string, map<string, CellData>> cellData;

    map<string, int> columnTotal;

    map<string, int> rowTotal;

    readData("inputdata.csv", cellData, columnTotal, rowTotal);

    writeResult("outputdata.csv", cellData, columnTotal, rowTotal);

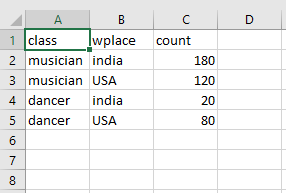
    cout << "Processing complete. Results saved to 't-d-weight-output.csv'." << endl;

    return 0;

}

**Result:**

**Input dataset:**



**Output:**

