```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <math.h>

using namespace std;

struct TypeFriend{
    string name;
    double Ne;
    double Ni;
    double Te;
    double Ti;
    double Se;
    double Si;
    double Fe;
    double Fi;
    string type;
};


int main()
{
    ifstream inFile;
    inFile.open("mbti.csv");
    string line;
    vector<string> lines;
    while (getline(inFile, line)){
        lines.push_back(line);
    }
    inFile.close();

    //push data to struct TypeFriend and split the data by comma
    vector<TypeFriend> types;
    for (int i = 0; i < lines.size(); i++){
        TypeFriend temp;
        stringstream ss(lines[i]);
        string item;
        getline(ss, item, ',');
        temp.name = item;
        getline(ss, item, ',');
        temp.Ne = stod(item);
        getline(ss, item, ',');
        temp.Ni = stod(item);
```

```
        getline(ss, item, ',');
        temp.Te = stod(item);
        getline(ss, item, ',');
        temp.Ti = stod(item);
        getline(ss, item, ',');
        temp.Se = stod(item);
        getline(ss, item, ',');
        temp.Si = stod(item);
        getline(ss, item, ',');
        temp.Fe = stod(item);
        getline(ss, item, ',');
        temp.Fi = stod(item);
        getline(ss, item, ',');
        temp.type = item;
        types.push_back(temp);
    }
    // calculate the distance between each friend and the user

    vector<double> arr;
    for (int i = 0; i < types.size(); i++){
        double temp = sqrt(pow(types[i].Ne - types[0].Ne, 2) +
pow(types[i].Ni - types[0].Ni, 2) + pow(types[i].Te - types[0].Te, 2)
+ pow(types[i].Ti - types[0].Ti, 2) + pow(types[i].Se - types[0].Se,
2) + pow(types[i].Si - types[0].Si, 2) + pow(types[i].Fe -
types[0].Fe, 2) + pow(types[i].Fi - types[0].Fi, 2));
        arr.push_back(temp);
    }

    // find the closest 3 friends and print their names
    string type1 = "" , type2 = "", type3 = "";
    int min1 = 1, min2 = 1, min3 = 1;
    for (int i = 1; i < arr.size(); i++){
        if (arr[i] < arr[min1]){
            min3 = min2;
            min2 = min1;
            min1 = i;
        }
        else if (arr[i] < arr[min2]){
            min3 = min2;
            min2 = i;
        }
        else if (arr[i] < arr[min3]){
            min3 = i;
        }
    }
    cout << types[min1].name << " is a " << types[min1].type << endl;
```

```
    cout << types[min2].name << " is a " << types[min2].type << endl;
    cout << types[min3].name << " is a " << types[min2].type << endl;



    //create my type from Typeout and print out my type
    string Typeout[3] = {types[min1].type, types[min2].type,
types[min3].type};
    string myType = "";
    for (int i = 0; i < 8; i++)
    {
        if (Typeout[0][i] == Typeout[1][i] and Typeout[0][i] ==
Typeout[2][i])
        {
            myType += Typeout[0][i];
        }
        else if (Typeout[0][i] == Typeout[1][i])
        {
            myType += Typeout[0][i];
        }
        else if (Typeout[0][i] == Typeout[2][i])
        {
            myType += Typeout[0][i];
        }
        else if (Typeout[1][i] == Typeout[2][i])
        {
            myType += Typeout[1][i];
        }
        else
        {
            myType += "x";
        }
    }
    cout << "your type is " << myType << endl;
    return 0;
}


// Language: cpp
// Path: mbti.csv
//Big(O) = O(n)
```

Output:

```
Pumipat Korncharornpisuit is a ISTJ
Thana Prajonkla is a ESTJ
Pakawat Kulchot is a ESTJ
your type is ESTJ
```