

```

#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

bool visitada(vector<int>& sol, int k, vector<vector<int>> const& mon, int& suma,
vector<int> const& pref) {

    int i = k - 1;

    bool valida = true;

    while (i >= 0 && i >= k - 2 && valida) {

        if (mon[sol[k]][sol[i]] == 1) {

            valida = false;
            suma -= pref[sol[k]];
        }
        else i--;
    }

    return valida;
}

bool esValida(vector<vector<int>> const& mon, vector<int>& sol, int& suma, int ruta, int
k, vector<bool> marca, vector<int> const& pref) {

    bool valida = true;

    if (marca[ruta]) {

        suma -= pref[ruta];
        valida = false;
    }

    return valida;
}

bool esSolucion(int k, int m) {
    return k == m - 1;
}

void tratarSolucion(vector<int>& mejor, int& maximo, vector<int>& sol, int& suma) {

    if (suma > maximo) {

        maximo = suma;
        mejor = sol;
    }
}

```

```

}

void mejorPlan(vector<vector<int>> const& mon, vector<int>& mejor, int& maximo,
vector<int>& sol, int& suma, int k, int n, int m, vector<int> const& pref, vector<bool>
marca) {

    for (int i = 0; i < n; i++) {

        sol[k] = i;
        suma += pref[i];
        if (esValida(mon, sol, suma, i, k, marca, pref) && visitada(sol, k, mon, suma,
pref)) {
            if (esSolucion(k, m)) {
                tratarSolucion(mejor, maximo, sol, suma);
            }
            else {
                marca[i] = true;
                mejorPlan(mon, mejor, maximo, sol, suma, k + 1, n, m, pref, marca);
                marca[i] = false;
            }
            suma -= pref[i];
        }
    }
}

void resuelveCaso() {

    int rutas, m;

    cin >> rutas >> m;

    vector<vector<int>> mon(rutas, vector<int>(rutas));

    int maximo = 0;
    int suma = 0;

    int p = 0;

    for (int i = 0; i < rutas; i++) {
        for (int j = 0; j < rutas; j++) {
            cin >> p;
            mon[i][j] = p;
        }
    }

    vector<int> preferencias(rutas);

    for (int i = 0; i < rutas; i++) {
        cin >> preferencias[i];
    }

    vector<int> sol(m, 0);
    vector<int> mejor(m, 0);
    vector<bool> marca(rutas, false);

```

```

mejorPlan(mon, mejor, maximo, sol, suma, 0, rutas, m, preferencias, marca);

if (maximo == 0) cout << "IMPOSIBLE" << endl;
else {
    cout << maximo << " ";

    for (int i = 0; i < m; i++) {
        cout << mejor[i] << " ";
    }

    cout << endl;
}
}

int main() {

    // Para la entrada por fichero.
#ifdef DOMJUDGE
    std::ifstream in("casos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    // Resolvemos
    int numCasos;
    std::cin >> numCasos;
    for (int i = 0; i < numCasos; ++i)
        resuelveCaso();

#ifdef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif

    return 0;
}

```