

```

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

using namespace std;

bool esValida(vector<int>& mer, int i) {

    return mer[i] < 3;
}

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

void tratarSolucion(int& minimo, int precios) {

    minimo = min(minimo, precios);
}

void mejorCompra(int& minimo, int precios, vector<vector<int>> const& ec, int k, int n,
int m, vector<int>& mer) {

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

        if (esValida(mer, i)) {
            precios += ec[i][k];
            mer[i]++;

            if (esSolucion(k, m)) {
                tratarSolucion(minimo, precios);
            }
            else {
                mejorCompra(minimo, precios, ec, k + 1, n, m, mer);
            }

            mer[i]--;
            precios -= ec[i][k];
        }
    }
}

void resuelveCaso() {

    int mercados, productos;

    cin >> mercados >> productos;

    vector<vector<int>> ec(mercados, vector<int>(productos));

    int minimo = 0;

```

```

    int p = 0;

    for (int i = 0; i < mercados; i++) {
        for (int j = 0; j < productos; j++) {
            cin >> p;
            ec[i][j] = p;
            minimo += p;
        }
    }

    int precios = 0;

    vector<int> centros(mercados, 0);

    mejorCompra(minimo, precios, ec, 0, mercados, productos, centros);

    cout << minimo << 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;
}

```