

```

NECESARIO "autoescuela.h"

#include <iostream>
#include <vector>
#include <fstream>

#include "autoescuela.h"

using namespace std;

bool resuelveCaso() {

    string comando, profesor, alumno;
    int nota;

    cin >> comando;

    if (!cin) {
        return false;
    }
    else {

        autoescuela autoescuela;

        while (comando != "FIN") {
            try {
                if (comando == "alta") {
                    cin >> alumno >> profesor;
                    autoescuela.alta(alumno, profesor);
                }
                else if (comando == "es_alumno") {
                    cin >> alumno >> profesor;
                    bool r = autoescuela.es_alumno(alumno, profesor);

                    if (r) {
                        cout << alumno << " es alumno de " << profesor << endl;
                    }
                    else {
                        cout << alumno << " no es alumno de " << profesor << endl;
                    }
                }
                else if (comando == "puntuacion") {
                    cin >> alumno;
                    int r = autoescuela.puntuacion(alumno);

                    cout << "Puntuacion de " << alumno << ": " << r << endl;
                }
                else if (comando == "actualizar") {
                    cin >> alumno >> nota;
                    autoescuela.actualizar(alumno, nota);
                }
                else if (comando == "examen") {
                    cin >> profesor >> nota;
                }
            }
            catch (...) {
                return false;
            }
        }
    }
}

```

```

        vector<string> resultado = autoescuela.examen(profesor, nota);

        cout << "Alumnos de " << profesor << " a examen:" << endl;

        for (size_t i = 0; i < resultado.size(); i++) {
            cout << resultado[i] << endl;
        }
    }
    else if (comando == "aprobar") {
        cin >> alumno;
        autoescuela.aprobar(alumno);
    }
}
catch (domain_error& e) {
    cout << "ERROR" << endl;
}

cin >> comando;
}

cout << "---" << endl;

return true;
}
}

```

```

int main() {
    // ajustes para que cin extraiga directamente de un fichero
#ifdef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    while (resuelveCaso());

    // para dejar todo como estaba al principio
#ifdef DOMJUDGE
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif
    return 0;
}

```