

```
NECESARIO "CarnetPuntos.h"
```

```
#include <iostream>
#include <fstream>
#include <algorithm>
```

```
#include "CarnetPuntos.h"
```

```
using namespace std;
```

```
bool resuelveCaso() {
```

```
    string comando, dni;
    int puntos;
```

```
    cin >> comando;
```

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

```
        carnetPuntos carnetPuntos;
```

```
        while (comando != "FIN") {
            try {
                if (comando == "nuevo") {
                    cin >> dni;
                    carnetPuntos.nuevo(dni);
                }
                else if (comando == "cuantos_con_puntos") {
                    cin >> puntos;
```

```
                    int resultado = carnetPuntos.cuantos_con_puntos(puntos); //lo pongo
asi para evitar bugs con las exceptions
```

```
                    cout << "Con " << puntos << " puntos hay " << resultado << endl;
```

```
                }
                else if (comando == "quitar") {
                    cin >> dni >> puntos;
                    carnetPuntos.quitar(dni, puntos);
                }
```

```
                else if (comando == "consultar") {
                    cin >> dni;
                    int resultado = carnetPuntos.consultar(dni);
                    cout << "Puntos de " << dni + ": " << resultado << endl;
                }
            }
```

```
        }
        catch (domain_error& e) {
            cout << "ERROR: " << e.what() << 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;
}

```