```cpp
#include <vector>
#include <string>
#include "QPanda.h"

using namespace std;
USING_QPANDA

QCircuit amplitude_encode(qvec q, vector<double> data);
QCircuit init_superposition_state(qvec q, size_t d);

class QSolver
{
public:
    QSolver(std::string cfg_file = "default.cfg");
    void run();
    vector<double> get_solution() { return m_solution; }
private:
    void init();
    QCircuit T_circuit_subspace(qvec qi, qvec qj, qvec qj_anc);
    QCircuit T_circuit(qvec qi, qvec qi_anc, qvec qj, qvec qj_anc);
    QCircuit W_circuit(qvec qi, qvec qi_anc, qvec qj, qvec qj_anc);
    QCircuit Chebyshev(size_t n, qvec qi, qvec qi_anc, qvec qj, qvec qj_anc);
    QCircuit Chebyshev_minus(size_t n, qvec qi, qvec qi_anc, qvec qj, qvec qj_anc);
    QCircuit one_iteration_qcir(qvec qt, qvec qi, qvec qi_anc, qvec qj, qvec
      qj_anc);
    vector<vector<size_t>> mapAllIndex(const vector<vector<size_t>>& data);
    double getSquareRoot(const vector<double>& data);
private:
    string m_cfg_file;
    size_t m_grid_number;
    size_t m_Cheby_times;
    size_t m_sparse_coef;
    double m_const_coef;
    vector<double> m_alpha;
    vector<double> m_sparse_matrix;
    vector<double> m_residual;
    vector<double> m_solution;
    vector<vector<size_t>> m_none_zero_block;
    vector<vector<size_t>> m_all_index_map;
};
```