```
#include <iostream>
#include <iomanip>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <assert.h>
#include <stdint.h>
using namespace std;
const int INIT = 1;     //initialized
const int HAVELABEL = 2;        //tensor with label
const int HAVEELEM = 4;     //tensor with elements
const int DISPOSABLE = 8;       //The elements of tensor is disposable through 'clon
e', 'reshapeClone', 'reshapeElem'. The elements of disposable Tensor is read-only.
const int ELEMFREED = 16;       //the memory space of elements is freed
#define DOUBLE      double
#include "Qnum.h"
#include "Bond.h"
#include "Block.h"
class Qnum_t;
class Block_t;
class Bond_t;

class SyTensor_t{
    public:
        SyTensor_t(vector<Bond_t>& _bonds, const string& _name = "Tensor");
        void reshape();
        void addLabel();
        void addRawElem();
        void contract();
        friend ostream& operator<< (ostream& os, SyTensor_t& SyT);
    private:
        string name;
        int status;         //Check initialization, 1 initialized, 3 initialized wit
h label, 5 initialized with elements
        vector<Bond_t> bonds;
        map<Qnum_t, Block_t> blocks;
        vector<int>labels;
        DOUBLE *elem;       //Array of elements
        int RBondNum;   //Row bond number
        int64_t elemNum;
        vector<Block_t*> RQidx2Blk;
        vector<bool> Qidx;
        vector<int> RQidx2Off;
        vector<int> CQidx2Off;
        static int counter;
        static int MEM;
        //Private Functions
        void grouping();
};
```