A lab uses 4 reagents and performs 2 different testes. The first test uses 3 units of the reagente R1, 1 of the reagent R2 and 2 of the reagent R4. The second test uses 1 unit of R1, 5 of R3, and 3 of R4. Use an enumeration, named *Reagente*, to define the constants R1, R2, R3, and R4.

Write a class named *Stock* to stock the quantities available of reagente. The class should contain methods to return the stock, increase and decrease it. You should also write input and output operators.

Write a class named *Lab* to run the stocks and tests. This class should contain a private member vector<Stock> to store the quantities available to each of the 4 reagents. The class should read the existencies from a file and store them when the class is destroyed (for this, you should use the class destructor).

The class for error treatment should be derived from *runtime_error.*

The main function should be:

```
int main()
try{
    Lab l("stocks.txt");
    cout << "Stock inicial:\n";
    l.mostrarStock();
    cout << "Comprar reagentes (s/n): ";
    char r;
    cin >> r;
    if(r==' s'  || r==' S' ) {
        l.compra(Reagente::R1,8);
        l.compra(Reagente::R2,8);
        l.compra(Reagente::R3,8);
        l.compra(Reagente::R4,8);

    }
    cout << "fazendo o teste1\n";
    l.teste1();
    cout << "fazendo o teste2\n";
    l.teste2();
    cout << "Stock final:\n";
    l.mostrarStock();
    return 0;
}
catch(const errLab& e) {
    cerr << e.what() << ' \n' ;
    return 1;
}
catch(...) {
    cerr << "Erro inesperado\n";
    return 2;
}
```

EXAMPLE FIRST EXECUTION

```
stocks.txt ainda não existe
Stock inicial:
R1: 0
R2: 0
R3: 0
R4: 0
Comprar reagentes (s/n): s
```

```
fazendo o teste1
fazendo o teste2
Stock final:
R1: 4
R2: 7
R3: 3
R4: 3
```

EXAMPLE SECOND EXECUTION

```
Stock inicial:
R1: 4
R2: 7
R3: 3
R4: 3
Comprar reagentes (s/n): n
fazendo o teste1
fazendo o teste2
Stock insuficiente
```