ksaveljev / **UVa-online-judge**

Branch: master ▾ UVa-online-judge / **394.cpp**

ksaveljev Initial commit: collection of my solutions to UVa problems 27c7d6e on Apr 4 2011

1 contributor

86 lines (69 sloc) 1.66 KB

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
using namespace std;

struct array {
        int base_address;
        int array_element_size;
        int d;
        int *c;
        int c0;
        vector< pair<int,int> > bounds;
        friend istream& operator>>(istream& istr, array& arr);
//      ~array();
};

// bad design of the solution leads to memory leaking ;)

//array::~array() {
//      delete [] c;
//}

istream& operator>>(istream& istr, array& arr) {
        int a,b;
        istr >> arr.base_address >> arr.array_element_size >> arr.d;
        for (int i = 0; i < arr.d; i++) {
                istr >> a >> b;
                //cout << "a = " << a << " b = " << b << endl;
                arr.bounds.push_back(make_pair(a,b));
        }
        arr.c = new int[arr.d];

        arr.c[arr.d - 1] = arr.array_element_size;

        for (int i = arr.d - 2; i >= 0; i--) {
                arr.c[i] = arr.c[i+1] * (arr.bounds[i+1].second - arr.bounds[i+1].first + 1);
        }

        arr.c0 = arr.base_address;
        for (int i = 0; i < arr.d; i++) {
                //cout << "arr.c[" << i << "] = " << arr.c[i] << endl;
                //cout << "arr.bound[" << i << "].first = " << arr.bounds[i].first << endl;
                //cout << "arr.bounds.size() = " << arr.bounds.size() << endl;
                arr.c0 -= arr.c[i] * arr.bounds[i].first;
        }
        //cout << "arr c0 = " << arr.c0 << endl;
}

int main(void) {
        int n, r, index, counter;
        string name;
        map<string,array> m;

        cin >> n >> r;

        while (n--) {
                array tmp;
                cin >> name;
                cin >> tmp;
                m[name] = tmp;
        }

        while (r--) {
                counter = 0;
                cin >> name;

                cout << name << "[";

                int result = m[name].c0;

                for (int i = 0; i < m[name].d; i++) {
                        cin >> index;
                        if (counter > 0)
                                cout << ", ";
                        cout << index;
                        result += m[name].c[i] * index;
                        counter = 1;
                }

                cout << "] = " << result << endl;
        }

        return 0;
}
```