

```

// BJU Bitwise Blitz
// prob3.cpp

#include <bits/stdc++.h>
using namespace std;

typedef long long ll;
typedef vector<int> vi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<vi> T_AdjList;
typedef vector<vii> T_WAdjList;

#define INF 1000000000
#define MOD 1000000007
#define LOne(S) (S & (-S))

class FenwickTree {
private:
    vi ft;
public:
    FenwickTree() {}
    FenwickTree(int n) { ft.assign(n+1, 0); }
    int rsq(int b) {
        int sum = 0; for (; b; b -= LOne(b)) sum += ft[b];
        return sum;
    }
    int rsq(int a, int b) {
        return rsq(b) - (a == 1 ? 0 : rsq(a-1));
    }

    void adjust(int k, int v) {
        for (; k < (int)ft.size(); k += LOne(k)) ft[k] += v;
    }
};

int main() {
    ios::sync_with_stdio(false);
    // code goes here
    int t; cin >> t;
    while (t--) {
        int roles, q, r; cin >> roles >> q >> r;
        set<int> your_roles;
        FenwickTree online(roles+1);
        for (int i = 0; i < r; ++i) {
            int x; cin >> x;
            your_roles.insert(x);
        }
        your_roles.insert(roles+1);
        for (int j = 0; j < q; ++j) {
            char type; int x, y = -1;
            cin >> type >> x;
            switch(type) {
                case 'A':
                    cin >> y;
                    online.adjust(x+1, y);
                    break;
                case 'B':
                    cin >> y;
                    online.adjust(x+1, -y);
                    break;
                case 'C':
                    your_roles.insert(x);
                    break;
            }
        }
    }
}

```