# C7I 数列询问

## 题目描述

给定两个数列 $\{a_i\},\{b_i\}$，多组询问，求 $\sum_{i+j=k} a_i \times b_j$

## 解题分析

从题目叙述可以看出比较模板，完全符合卷积的定义，但要注意构造时多项式从一次方项开始标号更方便计算

注意，由于 $1 \le k \le 2 \times 10^5$，故需要判断 $k$ 是否在卷积数组下标范围内

另外，注意 $-0$ 的输出，因为最后输出是整数，所以可以通过 (int)(x + 0.5) 来进行输出

## 代码示例

```
#include <bits/stdc++.h>

#define vc vector
#define vi vector<int>
#define vll vector<long long>
#define pii pair<int, int>
#define pll pair<long long, long long>
#define mod 998244353
#define db long double
#define ll long long
#define sz(x) ((int)(x.size()))
#define rep(i, a, b) for(int i = a; i <= b; i++)
#define per(i, a, b) for(int i = a; i >= b; i--)

using namespace std;

const db PI = acos((db)(-1));
const int maxn = 1e5 + 5, maxm = 2e5 + 5;

struct Complex {
    db x, y;

    Complex(db _x = 0.0, db _y = 0.0) { x = _x, y = _y; }

    Complex operator-(const Complex &b) const { return Complex(x - b.x, y - b.y); }
    Complex operator+(const Complex &b) const { return Complex(x + b.x, y + b.y); }
    Complex operator*(const Complex &b) const { return
Complex(x * b.x - y * b.y, x * b.y + y * b.x); }
};

void change(vc<Complex> &y, int n) {
    for(int i = 1, j = n / 2; i < n - 1; i++) {
        if(i < j) swap(y[i], y[j]);
        int k = n / 2;
        while(j >= k) { j -= k; k >>= 1; }
        if(j < k) j += k;
    }
}

void FFT(vc<Complex> &y, int n, int on = 1) {
    change(y, n);
    for(int h = 2; h <= n; h <<= 1) {
        Complex wn(cos(2 * PI / h), sin(on * 2 * PI / h));
        for(int j = 0; j < n; j += h) {
            Complex w(1, 0);
            for(int k = j; k < j + h / 2; k++, w = w * wn)
{
                Complex u = y[k], t = w * y[k + h / 2];
                y[k] = u + t;
                y[k + h / 2] = u - t;
            }
        }
    }
    if(on == -1)
        rep(i, 0, n - 1) y[i].x /= n;
}

void extend(vc<Complex> &a, int n, vc<Complex> &b, int m,
int &lim) {
    lim = 1;
    while(lim <= n + m) { lim <<= 1; }
    a.resize(lim, Complex());
    b.resize(lim, Complex());
}


inline void SOLVE() {

    int n, m, q, u;
    cin >> n >> m >> q;
    vc<Complex> a(n), b(m);
    rep(i, 0, n - 1) {
        cin >> u;
        a[i] = Complex(u, 0);
    }
    rep(i, 0, m - 1) {
        cin >> u;
        b[i] = Complex(u, 0);
    }

    int lim = 0;
    extend(a, n, b, m, lim);
    FFT(a, lim, 1); FFT(b, lim, 1);
    rep(i, 0, lim - 1) a[i] = a[i] * b[i];
    FFT(a, lim, -1);
    db t = (db)(0.5);
    while(q--) {
        int k; cin >> k; k -= 2;
        if(k < 0) cout << 0 << endl;
        else if(k >= lim) cout << 0 << endl;
        else {
            if(a[k].x - t < 1e-5) cout << 0 << endl;
            else cout << (int)(round(a[k].x)) << endl;
        }
    }

}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    int tt = 1;
    // cin >> tt;
    while(tt--) {
        SOLVE();
    }

    return 0;
}
```