**//http://codeforces.com/gym/100783/attachments/download/3773/20142015-acmicpc-southwestern-europe-regional-contest-swerc-14-en.pdf**

**//给你n个数，然后再给你一个数k,问这个数是否就是那n个数中的一个，或**

**//者说这个数可以由这n个数中的两个构成（可以是自己\*2）**

**/// 第 70 行 为重要代码**

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

**using namespace std;**

**const double PI = acos( -1);**

**#define FOR(i, a, b) for (int i = (a); i < (b); i++)**

**typedef double T;**

**const int MAXN = 1 << 20;**

**struct cplex {**

**T r, i;**

**cplex() : r(0), i(0) {}**

**cplex(T n) : r(n), i(0) {}**

**cplex(T r, T i) : r(r), i(i) {}**

**T real() {return r;}**

**cplex operator + (cplex b) {return cplex (r + b.r, i + b.i);}**

**void operator += (cplex b) {r += b.r, i += b.i;}**

**cplex operator - (cplex b) {return cplex(r - b.r, i - b.i);}**

**cplex operator \* (cplex b) {return cplex(r \* b.r - i \* b.i, r \* b.i + i \* b.r);}**

**cplex operator \*= (cplex b) {T r2 = r \* b.r - i \* b.i, i2 = r \* b.i + i \* b.r; r = r2, i = i2;}**

**void operator /= (T n) {r /= n, i /= n;}**

**};**

**cplex fa[MAXN << 1], fb[MAXN << 1];**

**void fft(cplex a[], int n, int invert) {**

**for (int i = 1, j = 0; i < n; i++) {**

**for (int s = n; j ^= s >>= 1, ~j & s;);**

**if (i < j) swap(a[i], a[j]);**

**}**

**for (int m = 1; m < n; m <<= 1) {**

**T p = PI / m \* (invert ? -1 : 1);**

**cplex w = cplex(cos(p), sin(p));**

**for (int i = 0; i < n; i += m << 1) {**

**cplex unit = 1;**

**for (int j = 0; j < m; j++) {**

**cplex& x = a[i + j + m], &y = a[i + j], t = unit \* x;**

**x = y - t;**

**y = y + t;**

**unit \*= w;**

**}**

**}**

**}**

**if (invert) for (int i = 0; i < n; i++) a[i] /= n;**

**}**

**void multiply(int a[], int b[], long long c[], int na, int nb) {**

**int n = 1; while (n < na + nb) n <<= 1;**

**for (int i = 0; i < n; i++) fa[i] = fb[i] = cplex(0);**

**for (int i = 0; i < na; i++) fa[i] = cplex(a[i]);**

**for (int i = 0; i < nb; i++) fb[i] = cplex(b[i]);**

**fft(fa, n, 0); fft(fb, n, 0);**

**for (int i = 0; i < n; i++) fa[i] \*= fb[i];**

**fft(fa, n, 1);**

**for (int i = 0; i < n; i++) c[i] = (long long) (fa[i].real() + 0.5);**

**}**

**const int maxn = 200000 + 10;**

**int n, m;**

**int a[MAXN];**

**int aa[MAXN];**

**long long b[MAXN << 1];**

**void solve() { // 计算 n \* m**

**int mid;**

**cin >> n >> m;**

**{**

**a[0] = n;**

**aa[0] = m;**

**multiply(a, aa, b, 10, 10);**

**/// 传入的参数为 a + a -> 可加到/可达到的 b**

**cout << b[0] << ' ';**

**}**

**}**

**void solve() {**

**// 计算**(1+2*x*)⋅(1+2*x*+*x*2)=1+4*x*+5*x*2+2*x*3(1+2x)⋅(1+2x+x2)=1+4x+5x2+2x3。

**int mid;**

**cin >> n >> m;**

**{**

**for(int i = 0; i <= n; i++)**

**cin >> mid, a[i] = mid;**

**for(int j = 0; j <= m; j++)**

**cin >> mid, aa[j] = mid;**

**multiply(a, aa, b, maxn, maxn);**

**/// 传入的参数为 a + a -> 可加到/可达到的 b**

**/// (1 + x + x^3 + x^5 ) \* (1 + x + x^3 + x^5 ) = (1 + x + x^2 + x^4 + x^6**

**/// x^6 + x^8 + x^10 )**

**for(int d = 0; d <= n + m; d++)**

**cout << b[d] << ' ';**

**}**

**void solve() { // 计算 1 3 5 + 2 4 5 7 8 9**

**cin >> n;**

**a[0] = 1;**

**FOR(i, 0, n) {**

**int k; cin >> k;**

**a[k] = 1;**

**}**

**multiply(a, a, b, maxn, maxn);**

**/// 传入的参数为 a + a -> 可加到/可达到的 b**

**cin >> m;**

**int ans = 0;**

**while (m--) {**

**int d; cin >> d;**

**if (b[d]) ans++;**

**}**

**cout << ans << "\n";**

**}**

**int main() {**

**solve();**

**printf("\nTime elapsed: %dms", 1000 \* clock() / CLOCKS\_PER\_SEC);**

**return 0;**

**}**

**/\***

**3**

**1 3 5**

**6**

**2 4 5 7 8 9**

**可达到的系数**

**1 2 4 6 8 10**

**Time elapsed: 2814ms 4**

**\*/**