C1-B

C1-B
题目内容
题目描述
输入格式
输出格式
输入样例1
输出样例1
输入样例2
输出样例2
Hint
题解思路
参考代码

题目内容

题目描述

Zhoues 对《算法导论》这本书简直是爱不释手，当他翻到这本书的第 23 页准备暗中观察时，立马被 Horner 规则给吸引到了，于是他打算做一个基于 Horner 规则的一元多项式计算器，帮助他进行一类特殊的二元多项式。

具体来说，给定如下两个一元多项式：

$$\sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

$$\sum_{i=0}^{m} b_i y^i = b_0 + b_1 y + b_2 y^2 + \ldots + b_m y^m$$

定义如下的二元多项式：

$$f(x, y) = \left(\sum_{i=0}^{n} a_i x^i\right)\left(\sum_{i=0}^{m} b_i y^i\right)$$

你需要处理 q 次计算，第 i 次计算给定两个变量值 X_i 和 Y_i，你需要求解 $f(X\_i, Y\_i) \bmod 10\,007$。

输入格式

第一行一个正整数 n（$1 \le n \le 3 \times 10^4$），表示第一个一元多项式的次数。

第二行 $n+1$ 个非负整数 $a\_0, a\_1, \ldots, a\_n$（$0 \le a_i \le 10^3$），表示第一个一元多项式的系数。

第三行一个正整数 m（$1 \le m \le 3 \times 10^4$），表示第二个一元多项式的次数。

第四行 $m+1$ 个非负整数 $b\_0, b\_1, \ldots, b\_n$（$0 \le b_i \le 10^3$），表示第二个一元多项式的系数。

第五行一个正整数 q（$1 \le q \le 2.5 \times 10^3$），表示计算的次数。

接下来 q 行，第 i 行两个非负整数 X_i, Y_i（$0 \leq X\_i, Y\_i \leq 10^4$），表示第 i 次计算给定的两个变量值。

输出格式

输出 q 行，第 i 行一个非负整数，表示 $f(X\_i, Y\_i) \bmod 10\,007$。

输入样例1

```
3
0 1 2 3
3
1 2 3 4
1
2 2
```

输出样例1

```
1666
```

输入样例2

```
5
100 200 300 400 900 1000
4
100 200 400 900 1000
5
1000 2000
2000 3000
4000 5000
6000 7000
8000 9000
```

输出样例2

```
7593
2016
3280
1949
2077
```

Hint

$ab \bmod c = (a \bmod c)(b \bmod c) \bmod c$

可以参考《算法导论》第二章课后习题 2-3。

题解思路

　该题目的灵感来源于《算法导论》第二章课后习题 2-3，对于一元多项式的计算提供了一个很好的解决思路，即对于

$$\sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

我们可以先分离常数项与带有x这个变量的部分，并对包含x的部分继续提取x，使该部分重新分为常数项和带有x的部分，重复上述步骤，直到最后无法提取x为止。

$$a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = a_0 + x(a_1 + x(a_2 + \ldots x(a_{n-1} + a_n x)))$$

我们观察上面的公式不难发现，其实我们是在不断的重复计算$(a_{n-1} + a_n x)$，并在计算的结果上乘以x并加上a_{n-2}，不断重复该计算最后就可以得到结果。

因此可以使用递推或者递归完成该代码的编写，参考代码采用的是递归的方式来进行编写的，递推同理

计算一组 $f(X_i, Y_i)$ 的时间复杂度为 $O(n+m)$，总时间复杂度为 $O(q(n+m))$。

参考代码

```
#include<stdio.h>
#define M 10007
long long a[1000001],b[1000001];
long long n,m,x,y,i,t,q;
long long Horner(long long *a,long long n,long long x) {
    if(n<1)
        return a[0] % M;
    else
        return (a[0] % M + (Horner(a+1,n-1,x) * x) % M) % M;
}

int main() {
    scanf("%lld",&n);
    for(i=0; i<=n; i++) {
        scanf("%lld",&a[i]);
    }
    scanf("%lld",&m);
    for(i=0; i<=m; i++) {
        scanf("%lld",&b[i]);
    }
    scanf("%lld",&q);
    while(q--)
    {
        scanf("%lld %lld",&x,&y);
        printf("%lld\n",( (Horner(a,n,x) % M) * (Horner(b,m,y) % M) ) % M );
    }
    return 0;
}
```

Author：周恩申