C：

没有

C++：

#include <iostream>

#include <fstream>

#include <cstdio>

#include <cstring>

#include <algorithm>

#include <queue>

#include <stack>

#include <climits>

#include <ctime>

#include <cmath>

#include <set>

#include <map>

#include <string>

#include <vector>

#define MAX(a,b) ((a)>(b)?(a):(b))

#define MIN(a,b) ((a)<(b)?(a):(b))

#define abs(x) ((x)>0?(x):(-(x)))

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

#define FORD(i,a,b) for(int i = (a);i>=(b);i--)

#define REP(i,n) for(int i = 0;i<(n);i++)

#define rst(x,k) memset(x,k,sizeof(x))

#define lowbit(x) ((x)&(-(x)))

//#define h(x) (1<<(x))

//#define lson (ind<<1)

//#define rson (ind<<1|1)

#define eps 1e-6

#define INF 140000000

#define maxn 5000

#define mod 1000000007

#define Pi acos(-1.0)

#define link fjksldfjaslkdfjas

using namespace std;

typedef long long LL;

int a[maxn] , b[maxn];

int visit[maxn];

char ss[maxn];

struct node{

int t,nxt;

}edge[maxn << 2];

int headline[maxn] , E , n , tot;

void add(int f,int t){

edge[E].t = t;

edge[E].nxt = headline[f];

headline[f] = E++;

}

bool dfs(int u){

bool ok = true;

visit[u] = true;

for(int i = headline[u];~i;i = edge[i].nxt){

int v = edge[i].t;

if(!visit[v]){

if(dfs(v)){

ok = false;

break;

}

}

}

visit[u] = false;

if(ok == false)return false;

return true;

}

void solve(void){

n = 0; tot = 0;

int len = strlen(ss) , temp = 0;

rst(headline,-1); E = 0; rst(visit,false);

bool finish = true;

REP(i,len){

if(ss[i] <= '9' && ss[i] >= '0'){

temp \*= 10;

temp += ss[i] - '0';

finish = false;

}else if(!finish){

finish = true;

n++; a[n] = temp;

temp = 0;

}

}

if(!finish){

n++; a[n] = temp;

finish = true;

temp = 0;

}

gets(ss);

len = strlen(ss);

REP(i,len){

if(ss[i] <= '9' && ss[i] >= '0'){

temp \*= 10;

temp += ss[i] - '0';

finish = false;

}else if(!finish){

finish = true;

tot++; b[tot] = temp;

temp = 0;

}

}

if(!finish){

tot++; b[tot] = temp;

finish = true;

temp = 0;

}

sort(a+1,a+n+1);

sort(b+1,b+tot+1);

//FOR(i,1,n)printf("%d ",a[i]);printf("\n");

//FOR(i,1,tot)printf("%d ",b[i]);printf("\n");

FOR(i,1,n){

FOR(j,i+1,n){

if(a[i] % a[j] == 0 || a[j] % a[i] == 0){

add(i,j); add(j,i);

}

}

}

FOR(i,1,tot){

FOR(j,1,n){

if(b[i] == a[j]){

if(dfs(j)){

printf("%d\n",a[j]);

return;

}

break;

}

}

}

printf("-1\n");

}

int main(void){

while(gets(ss)) solve();

return 0;

}