Nguyễn Xuân Huy

Viện CNTT Viện HL KH & CN VN

0903203800 nxhuy564@gmail.com

**ADVANCED ALGORITHMS AND APPLICATIONS**

**CONTEST 2**

**PTIT 2023**

Contents

[Contest 2 3](#_Toc138957958)

[XÂU AB 3](#_Toc138957959)

[TẬP QUÂN SỰ 3](#_Toc138957960)

[HOÁN VỊ KẾ TIẾP 3](#_Toc138957961)

[DÃY XÂU FIBONACCI 3](#_Toc138957962)

[NGƯỜI DU LỊCH 3](#_Toc138957963)

[CHIA BÁNH 3](#_Toc138957964)

[HÀNH TRÌNH ĐẶC BIỆT 4](#_Toc138957965)

[ƯỚC SỐ CHUNG CỦA DÃY SỐ 4](#_Toc138957966)

[HỆ THỐNG NHẬN DẠNG 4](#_Toc138957967)

[SỐ NGUYÊN TỐ LỚN NHẤT 5](#_Toc138957968)

[Bài giải Contest 2 5](#_Toc138957969)

[XÂU AB 5](#_Toc138957970)

[TẬP QUÂN SỰ 6](#_Toc138957971)

[HOÁN VỊ KẾ TIẾP 8](#_Toc138957972)

[DÃY XÂU FIBONACCI 9](#_Toc138957973)

[NGƯỜI DU LỊCH 10](#_Toc138957974)

[ƯỚC SỐ CHUNG CỦA DÃY SỐ 11](#_Toc138957975)

[SỐ NGUYÊN TỐ LỚN NHẤT 12](#_Toc138957976)

## Contest 2

### XÂU AB

Một xâu kí tự S được gọi là xâu AB độ dài n nếu mọi kí tự trong S đều có giá trị A hoặc B.

Ví dụ xâu S = “ABABABAB” là một xâu AB độ dài 8.

Với mỗi cặp giá trị n và k, hãy liệt kê các xâu AB dài n và chứa duy nhất một dãy k kí tợ A liên tiếp.

Ví dụ

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 5 3 | 5  AAABA  AAABB  ABAAA  BAAAB  BBAAA |

### TẬP QUÂN SỰ

Tại Chương Mỹ Resort, vào nửa đêm, cả trung đội nhận lệnh tập trung ở sân.

Mỗi chiến sỹ được đánh số từ 1 đến N (1 < N < 40).

Giám thị yêu cầu chọn ra một dãy K chiến sỹ để tập đội ngũ và cứ lần lượt duyệt hết

tất cả các khả năng chọn K người như vậy từ nhỏ đến lớn (theo số thứ tự).

Bài toán đặt ra là cho một nhóm K chiến sỹ hiện đang phải tập đội ngũ,

hãy tính xem trong lượt chọn K người tiếp theo thì mấy người trong nhóm cũ sẽ được tạm nghỉ.

Nếu đã là nhóm cuối cùng thì tất cả đều sẽ được nghỉ.

### HOÁN VỊ KẾ TIẾP

Cho số tự nhiên x dài tối đa 2000 chữ số. Hãy hoán vị các chữ số của x để thu đươc số sát sau của x,

tức là tìm số tự nhiên y > x và giữa x và y không tồn tại số z thu được bằng cách hoán vị các chữ số của x.

Ví dụ: x = 2092731; y = 2093127.

### STRING FIBONACCI

Một string G chỉ bao gồm các chữ cái A và B được gọi là string Fibonacci nếu thỏa mãn tính chất:

G(1) = A; G(2) = B; G(n) = G(n-2)+G(n-1), n > 2, và phép cộng (+) là phép nối hai xâu với nhau.

Bài toán đặt ra là tìm ký tự ở vị trí thứ k (tính từ 1) của string Fibonacci thứ n.

### NGƯỜI DU LỊCH

Cho n thành phố đánh số từ 1 đến n (n <= 15) và các tuyến đường giao thông hai chiều giữa chúng,

mạng lưới giao thông này được cho bởi mảng C[1…n, 1…n] ở đây C[i][j] = C[j][i] là chi phí đi đoạn đường

trực tiếp từ thành phố i đến thành phố j. Một người du lịch xuất phát từ thành phố 1, muốn đi thăm

tất cả các thành phố còn lại mỗi thành phố đúng 1 lần và cuối cùng quay lại thành phố 1.

Hãy chỉ ra chi phí ít nhất mà người đó phải bỏ ra.

|  |  |
| --- | --- |
| TOUR.INP | OUTPUT |
| 4  0 20 35 10  20 0 90 50  35 90 0 12  10 50 12 0 | 117 |

### CHIA BÁNH

Kết thúc buổi thuyết trình, cô giáo thưởng cho các nhóm sinh viên bằng bánh pizza.

Mỗi nhóm sau khi báo cáo xong sẽ nhận được bánh pizza từ cô giáo.

Có tất cả N nhóm sinh viên, mỗi nhóm gồm có A[i] bạn. Mỗi chiếc bánh pizza có P miếng bánh.

Cô giáo sẽ lần lượt mở từng hộp bánh chia cho các bạn sinh viên, và luôn duy trì lúc nào cũng chỉ có

nhiều nhất 1 chiếc hộp bánh đang còn các miếng bánh thừa. Miếng bánh nào thừa từ nhóm sinh viên

trước sẽ được chia cho nhóm sinh viên sau, khi hết các miếng bánh thừa, các hộp bánh mới sẽ được mở tiếp.

Ví dụ 4 nhóm sinh viên có 5, 6, 4, 4 bạn, và mỗi chiếc bánh có 3 miếng. Cô giáo sẽ phải mở 2 hộp bánh

để chia cho nhóm sinh viên đầu tiên, và còn dư 1 miếng bánh. Miếng bánh này sẽ được phát cho một bạn

sinh viên ở nhóm 2, sau đó cô giáo sẽ phải mở tiếp 2 hộp bánh mới, và vẫn dư lại 1 miếng bánh.

Nhận thấy rằng việc có các miếng bánh bị thừa sau khi chia là không hay, vì vậy cô giáo muốn sắp xếp các nhóM

sinh viên cho hợp lý, sao cho số lượng nhóm sinh viên nhận được toàn miếng bánh từ hộp bánh mới là

nhiều nhất.

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 3  4 3  4 5 6 4  4 2  4 5 6 4  3 3  1 1 | 3  4  1 |

### HÀNH TRÌNH ĐẶC BIỆT

Trong giấc mơ, Tí thấy mình đang ở trong một vương quốc cổ đẹp với rất nhiều những hòn đảo và cây cầu.

Tí muốn đi tham quan tất cả các hòn đảo với chi phí nhỏ nhất có thể. Đang bước đi dạo quanh con sông,

Tí tình cờ gặp Tèo và cả hai cùng bắt đầu hành trình. Tuy nhiên, Tèo muốn sắp xếp một hành trình hợp lý,

sao cho mỗi hòn đảo chỉ đến thăm duy nhất một lần, và trước khi đến thăm hòn đảo thứ K, thì tất cả các

hòn đảo có nhãn nhỏ hơn K đã phải được thăm rồi hoặc sẽ được thăm sau đó.

Các bạn hãy giúp Tí và Tèo xây dựng dựng hành trình du lịch tối ưu nhất!

Dữ liệu vào: Dòng đầu tiên là số nguyên dương N là số hòn đảo (1 <= N <= 1500).

N dòng tiếp theo, mỗi dòng gồm N số trong khoảng từ 0 đến 1000. Số ở hàng A, cột B thể hiện chi phí đi lại

giữa thành phố A và B (dĩ nhiên luôn bằng với số ở hàng B, cột A). Nếu A = B, AB = 0.

Kết quả: In ra một số nguyên dương duy nhất là chi phí nhỏ nhất cho hành trình.

Hành trình tối ưu của Test 1 là 2, 1, 3 hoặc 3, 1, 2. Hành trình 1, 3, 2 có chi phí nhỏ hơn nhưng không thỏa

mãn điều kiện mà Tèo đưa ra.

Giải thích test 2: Hành trình tối ưu là 3, 1, 2, 4 hoặc 4, 2, 1, 3.

|  |  |
| --- | --- |
| Test 1 | Test 2 |
| Input:  3  0 5 2  5 0 4  2 4 0  Output: 7 | Input:  4  0 15 7 8  15 0 16 9  7 16 0 12  8 9 12 0  Output: 31 |

### ƯỚC SỐ CHUNG CỦA DÃY SỐ

Cho trước một dãy số nguyên. Bạn được phép giảm đi mỗi số trong dãy một giá trị không vượt quá k.

Giờ với mọi thao tác thay đổi dãy, ta được một dãy mới. Hãy tìm ra ước số chung lớn nhất

có thể của dãy số đã cho.

|  |  |  |
| --- | --- | --- |
| GCD.INP | Output | Giải thích |
| 5 2  4 7 11 13 15 | 3 | Ucln(4-1, 7-1, 11-2, 13-1, 15)  = Ucln(3, 6, 9, 12, 15) = 3 |

### HỆ THỐNG NHẬN DẠNG

Một hệ thống nhận diện khuôn mặt gồm có N module. Mỗi module có khả năng hoạt động chính xác bằng P[i].

Xác suất hoạt động chính xác của hệ thống được xác định bằng tích của tất cả các module.

Để tăng độ chính xác của hệ thống, bạn phải thực hiện huấn luyện (train) dữ liệu cho mỗi module. Tuy nhiên,

việc này mất rất nhiều thời gian và bạn chỉ có tổng cộng U đơn vị thời gian. Train một model trong X đơn vị

thời gian, độ chính xác của module này tăng lên thêm X (tối đa là bằng 1). Bạn hãy xác định xem sau khi

training, độ chính xác lớn nhất mà hệ thống đạt được là bao nhiêu?

Dữ liệu vào: Dòng đầu tiên là số lượng bộ test T (1 <= T <= 100).

Mỗi test gồm số nguyên dương N (1 <= N <= 50).

Dòng tiếp theo là số thực U. Dòng cuối gồm N số thực P[i] (0 <= P[i] <= 1).

Kết quả: Với mỗi test in ra trên một dòng đáp án tìm được với độ chính xác 10^-6.

|  |  |
| --- | --- |
| Input | Output |
| 2  4  1.4000  0.5000 0.7000 0.8000 0.6000  2  1.0000  0.0000 0.0000 | 1.000000  0.250000 |

### SỐ NGUYÊN TỐ LỚN NHẤT

Cho một xâu các chữ số, nhiệm vụ của bạn là tìm kiếm số nguyên tố lớn nhất (gồm các chữ số liên tiếp nhau)

xuất hiện trong xâu. Các số nguyên tố tìm được sẽ ở trong đoạn 2 đến 100000.

|  |  |
| --- | --- |
| Input | Output |
| 11245  91321150448  1226406  0 | 11  1321  2 |

## Bài giải Contest 2

### XÂU AB

**Algorithm**

Bài này có dữ liệu nhỏ (n <= 30) nên có thể dùng phương pháp vét cạn (brute force).

Độ phức tạp: O(2^n)

Duyệt các dãy 0/1 dài n bit tong đó có đúng n bit 1 liên tiếp thì đưa vào kết quả.

1:A, 0:B

**Program**

/\*

Name: ABSTR.CPP (xau AB)

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

\*/

#include <bits/stdc++.h>

using namespace std;

int form = 0;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

// Hien thi x dang A(1)B(0)

string Str(int x, int n) {

string AB = "BA";

string s = "";

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

s = AB[x & 1] + s;

x >>= 1;

}

// cout << "\n " << s;

return s;

}

bool Test(int x, int n, int k) {

int d = 0;

while(x > 0) {

//cout << "\n Test x = " << x << ": " << Str(x,n);

if((x & form) == form) ++d;

x >>= 1;

}

//cout << "\n d = " << d; Go();

return d == 1;

}

int XauAB(int n, int k) {

/\*-------------------

n = 5; k = 3

minval = 00111

maxval = 11101

---------------------\*/

form = (1 << k) - 1;

cout << "\n form = " << form;

// min val la so min n bit chua dung k bit 1

int minval = form;

// max val la so max n bit chua dung k bit 1

int maxval = 1 << n;

cout << "\n minval = " << minval << " -> " << Str(minval,n);

cout << "\n maxval = " << maxval << " -> " << Str(maxval,n+1);

//Go();

vector<string> ss;

int d = 0;

for(int val = minval; val < maxval; ++val) {

if (Test(val, n, k)) {

++d;

// cout << "\n " << Str(val,n); //Go();

ss.push\_back(Str(val, n));

}

}

cout << "\n Total: " << d;

for(int i = 0; i < ss.size(); ++i)

cout << "\n " << ss[i];

}

main() {

XauAB(5, 2);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### TẬP QUÂN SỰ

**Algorithm**

* Biết tổ hợp chặp k của n phần tử là c.
* Tìm tổ hợp sát sau của c là c'
* Tìm phần giao của c và c'

Ví dụ, n = 5, k = 3 : 1 2 3 4 5

Ta có 10 các tổ hợp

1 2 3, 1 2 4, 1 2 5, 1 3 4, 1 3 5, 1 4 5, 2 3 4, 2 3 5, 2 4 5, 3 4 5.

Sau tổ hợp 1 3 5 là tổ hợp 1 4 5, Sẽ có 2 người được nghỉ là 1 và 5

Sau tổ hợp 3 4 5 không có tổ hợp nào, do đó cả 3 bạn 3, 4 và 5 đều được nghỉ.

**Độ phức tạp**

O(n)

**Program**

/\*

Name: TQS.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Tap quan su

\*/

#include <bits/stdc++.h>

using namespace std;

typedef vector<int> VI;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void Print(VI c, const char \* msg = "") {

cout << msg;

for(int i = 0; i < c.size(); ++i)

cout << " " << c[i];

}

// To hop sat sau cc cua TH c

VI Next(VI c, int n) {

int k = c.size();

cout << "\n k = " << k;

VI cc(c); // copy c -> cc

// cc.assign(c.begin(), c.end());

Print(c, "\n c: ");

//Print(cc, "\n cc: ");

// duyet nguoc, tim diem gay

int m = n;

int i = 0;

// cc = 1 3 4, m = 5

for(i = k - 1; i >= 0; --i, --m) {

// cout << "\n i = " << i << " cc[i] = " << cc[i] << " m = " << m;

if (cc[i] != m) break;

}

cout << "\n Diem gay i = " << i;

if (i < 0) return cc;

// Rai deu

int v = cc[i] + 1;

for(int j = i; j < k; ++j, ++v) cc[j] = v;

return cc;

}

VI Giao(VI c, VI cc, int n) {

++n;

char b[n];

fill(b, b+n, 0);

for(int i = 0; i < cc.size(); ++i) {

++b[cc[i]] = 1;

}

for(int i = 0; i < c.size(); ++i) {

++b[c[i]];

}

VI g;

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

if (b[i] == 2) g.push\_back(i);

}

return g;

}

void QS() {

// doc du lieu

ifstream f("QS.INP");

int n, k, q;

f >> n >> k >> q;

cout << "\n n = " << n << " k = " << k << " so test = " << q;;

VI c;

for(int t = 1; t <= q; ++t) {

cout << "\n\n Test No " << t << ": ";

c.clear();

for(int i = 0; i < k; ++i) {

int x;

f >> x;

c.push\_back(x);

cout << " " << x;

} // c

VI cc = Next(c, n) ;

Print(cc, "\n Next c -> cc = ");

Print(Giao(c,cc,n), "\n Result: ");

} // t

}

main() {

QS();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

/\* QD.INP

5 3 (n k)

3 (3 tést)

1 3 4

3 4 5

1 2 5

\*/

### HOÁN VỊ KẾ TIẾP

**Algorithm**

Two phases

* Duyệt ngược tìm điểm gảy i: a[i] < a[i+1]
* Nếu ko có điểm gãy (i < 0): return false
* Duyệt ngược tìm điểm vượt j: a[j] > a[i]
* Đổi chỗ a[i], a[j]
* Lật a[i+1..n-1]

**Complexity:** O(n)

**Program**

/\*

Name: NextNum.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

\*/

#include <bits/stdc++.h>

using namespace std;

typedef vector<int> VI;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

bool NextNum(string & s) {

int len = s.length();

int i, j;

for(i = len-2; i >= 0; --i)

if(s[i] < s[i+1]) break;

if(i < 0) return false;

for(j = len-1; j >= 0; --j)

if(s[j] > s[i]) break;

int x = s[i]; s[i] = s[j]; s[j] = x;

++i; j = len-1;

while(i < j) {

x = s[i]; s[i] = s[j]; s[j] = x;

++i; --j;

}

return true;

}

main() {

string s = "2002731";

cout << "\n Input s: " << s;

NextNum(s); // "2003127"

cout << "\n Output s: " << s;

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### DÃY XÂU FIBONACCI

**Algorithm**

---------------f(n-2)---------+-------------------f(n-1)-------------------------

**Complexity**: O(n)

**Program**

/\*

Name: STRFIB.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: String Fibonacci

\*/

#include <bits/stdc++.h>

using namespace std;

typedef long long Long;

typedef vector<Long> VL;

const int MN = 80; // nang thanh 1M

VL f; // primes

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

void GenFib() {

f.clear();

f.push\_back(0); f.push\_back(1); // f[0] = 0, f[1] = 1

for(int i = 2; i <= 80; ++i) {

f.push\_back(f[i-2]+f[i-1]);

// cout << "\n f[" << i << "] = " << f.back();

}

// cout << "\n " << f.back();

}

/\*-------------------------------------------------------------

n F

1 A

2 B

3 A|B

4 B|AB

5 AB|BAB

6 BAB|ABBAB

7 ABBAB|BABABBAB

8 BABABBAB|ABBABBABABBAB

9 ABBABBABABBAB|BABABBABABBABBABABBAB

10 BABABBABABBABBABABBAB|ABBABBABABBABBABABBABABBABBABABBAB

---------------------------------------------------------------\*/

char Find(Long n, Long k) {

//cout << "\n n = " << n << " f[n] = " << f[n] << " k = " << k;

while(n > 2) {

// f[n] = f[n-2] + f[n-1]

if(k > f[n-2]) {

k -= f[n-2];

n -= 1;

}

else n -= 2;

}

// cout << "\n The end: n = " << n << " k = " << k;

return (n == 1) ? 'A' : 'B';

}

void Test(Long n) {

for(int k = 0; k < f[n]; ++k)

cout << Find(n,k);

}

main() {

GenFib();

Test(10);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### NGƯỜI DU LỊCH

**Algorithm**

Dùng hàm Next duyệt mọi hoán vị.

**Complexity**: O(n!). Nếu ma trận đối xứng thì chỉ cần duyệt n!/2 lần

Mọi con đường đều có dạng 1 → x → 1, trong đó x là một hoán vị của 2..n.

n = 5; x có dạng

2345, 2354, 2435, 2453, 2534, 2543

3245, 3254, 3425, 3452, 3524, 3542

4235, 4253, 4325, 4352, 4523, 4532

5234, 5243, 5324, 5342, 5423, 5432

2345 và 5432 là đối xứng

**Program**

/\*

Name: TOURIST.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description:

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 16;

typedef int \* Pint; // Pointer to int

Pint \*c;

Pint p; // permutation

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

bool Next(int n) { // Next of p[2..n]

int i, j;

for(i = n-1; i > 1; --i)

if (p[i] < p[i+1]) break;

if(i < 2) return false;

for(j = n; j > i; --j)

if (p[j] > p[i]) break;

int x = p[i]; p[i] = p[j]; p[j] = x;

++i; j = n;

while(i < j) {

x = p[i]; p[i] = p[j]; p[j] = x;

++i; --j;

}

return true;

}

void Print(Pint a, int d, int c, const char \* msg = "") {

cout << msg;

for(int i = d; i <= c; ++i) cout << " " << a[i];

}

int Sum(int n) {

// 1-> 2 -> ... -> n -> 1

int v = c[p[n]][1];

for(int i = 2; i <= n; ++i)

v += c[p[i-1]][p[i]];

return v;

}

void Run() {

ifstream f("TOUR.INP");

int n;

f >> n;

cout << "\n n = " << n;

c = new Pint[n+1];

for(int i = 1; i <= n; ++i) c[i] = new int[n+1];

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

for(int j = 1; j <= n; ++j) {

f >> c[i][j];

} // j

} // i

f.close();

cout << "\n c: ";

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

cout << "\n ";

for(int j = 1; j <= n; ++j) {

cout << " " << c[i][j];

} // j

} // i

cout << "\n Init p: " ;

int i = 1;

p = new int[n+1]; // xuat phat tu TP 1

p[1] = 1;

for(int i = 2; i <= n; ++i) {

p[i] = i;

cout << " " << i;

}

int minval = Sum(n);

cout << "\n Init minval = " << minval;

// cac hoan vi 2..n

while(Next(n)) {

if(p[2] == n) break;

Print(p, 2, n, "\n"); //Go();

int val = Sum(n);

cout << " val = " << val;

minval = min(minval, val);

}

cout << "\n minval = " << minval;

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### ƯỚC SỐ CHUNG CỦA DÃY SỐ

**Algorithm**

* Ucln của dãy sô (nguyên không âm) không vượt quá trị min của dãy.
* Đọc dữ liệu n, k, a[0..n-1]; xác địnhj vmin

for v = vmin .. 0:

if all a[0..n-1] % v <= k:

lấy nghiệm

**Complexity:** O(n)

**Program**

/\*

Name: GCD.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Ucln cua day so

GCD.INP

5 2

4 7 11 13 15

\*/

#include <bits/stdc++.h>

using namespace std;

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

bool Test(int a[], int n, int k, int vmin) {

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

if(a[i] % vmin > k) return false;

return true;

}

void Run() {

int x, vmin;

ifstream f("GCD.INP");

int n, k;

f >> n >> k;

cout << "\n n = " << n << " k = " << k;

int a[n];

f >> vmin; a[0] = vmin;

cout << "\n a: " << a[0];

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

f >> a[i];

vmin = min(vmin, a[i]);

cout << " " << a[i];

}

cout << "\n vmin = " << vmin;

f.close();

for(; vmin > 0; --vmin)

if(Test(a, n, k, vmin)) {

cout << "\n Result: " << vmin;

return;

}

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### SỐ NGUYÊN TỐ LỚN NHẤT

**Algorithm**

Two phases

* Gọi Sieve(n) tìm các sô nguyên tố < n. Chuyển qua str ghi vào vector p
* Đọc từng số x trong input file x. Duyệt ngược p, tìm xuất hiện của p[i] trong x

**Complexity:** O(n)

**Program**

/\*

Name: PRIME.CPP

Copyright: (C) 2023

Author: DevCPP Fan

Date: 25-06-23 15:43

Description: Max prime

PRIME.INP

711245

1011031101

91321150448

1226406

87919999532022

556774593141

13392639103

2661705371673

0

Output

711245 -> 71

1011031101 -> 110311

91321150448 -> 1321

1226406 -> 2

87919999532022 -> 999953

556774593141 -> 774593

13392639103 -> 339263

2661705371673 -> 617053

\*/

#include <bits/stdc++.h>

using namespace std;

typedef vector<string> VI;

const int MN = 1000000; // nang thanh 1M

VI p; // primes

void Go() {

cout << " ? ";

fflush(stdin);

if(cin.get() == '.') exit(0);

}

string Str(int x) {

stringstream ss;

ss << x;

string s;

ss >> s;

return s;

}

// all primes < MN

void Sieve() {

bitset<MN> b;

b.set(); // all 1

b[0] = b[1] = 0; // 0, 1 ko nguyen to

// xoa cac boi cua 2 ke tu 4

for(int j = 4; j < MN; j += 2) b[j] = 0;

for(int i = 3; i\*i <= MN; i += 2) {

if(b[i]) { // i nguyen to

// xoa cac boi cua i ke tu i\*i

for(int j = i\*i; j < MN; j += i) {

b[j] = 0;

}

} // if

} // i

//for (int i = 1; i < 100; ++i)

//if(b[i]) cout << " " << i;

// ghi nguoc vao

p.clear();

p.push\_back(Str(2));

for(int i = 3; i < MN; i += 2) {

if(b[i]) p.push\_back(Str(i));

}

/\*

for(int i = 0; i < 30; ++i)

cout << "\n " << p[i];

\*/

}

// so ng to dau tien trong p

string OldMatch(string s) {

for(int i = p.size()-1; i >= 0; --i) {

size\_t found = s.find(p[i]);

if (found != string::npos)

return p[i];

}

return "";

}

string Match(string s) {

for(int i = p.size()-1; i >= 0; --i) {

if (s.find(p[i]) != string::npos)

return p[i];

}

return "";

}

void Run() {

Sieve();

ifstream f("PRIME.INP");

string s;

while(true) {

f >> s;

if (s == "0") break;

string v = Match(s);

if (v != "") cout << "\n " << s << " -> " << v;

}

f.close();

}

main() {

Run();

cout << endl << "\n\n T h e E n d \n";

return 0;

}

29 June 2023