

# XỬ LÝ SỐ NGUYÊN LỚN TRONG C++

## PHỤ LỤC

Phần thứ nhất	3
MỞ ĐẦU .....	3
Phần thứ hai .....	4
NỘI DUNG CHUYÊN ĐỀ.....	4
I. CÁC PHÉP TOÁN XỬ LÝ SỐ NGUYÊN LỚN .....	4
I.1. Biểu diễn số nguyên lớn.....	4
I.2. Các phép toán xử lý số nguyên lớn.....	4
I.2.1. Cộng 2 số nguyên lớn.....	4
I.2.2. Trừ 2 số nguyên lớn (Trừ số lớn cho số bé).....	5
I.2.3. Nhân một số nguyên lớn với một nguyên số nhỏ.....	6
I.2.4. Nhân 2 số nguyên lớn.....	7
I.2.5. Chia số nguyên lớn cho số nguyên nhỏ.....	8
I.2.6. Chia hai số nguyên lớn.....	9
II. BÀI TẬP VẬN DỤNG .....	11
II.1. Tách N thành tổng các số fibonacci (bài 2.11 sgk chuyên tin quyển 1) ..	11
II.2. Chuyển cơ số (bài 2.13 sgk chuyên tin quyển 1) .....	15
II.3. Nguồn của một số (bài 2.18 sgk chuyên tin quyển 1).....	19
II.4. Số các ước và tổng ước của N! (bài 2.19 sgk chuyên tin quyển 1) .....	22
II.5. Cân đĩa (Bài 2.20 sgk chuyên tin quyển 1).....	27
II.6. Hoán vị xâu (bài 2.22 sgk chuyên tin quyển 1) .....	32
II.7. Đánh số trang sách (bài 2.23 sgk chuyên tin quyển 1) .....	35
II.8. Hái nấm (bài 2.25 sgk chuyên tin quyển 1) .....	39
III. BÀI TẬP LUYỆN TẬP.....	44
Phần thứ ba .....	45
KẾT LUẬN .....	45
TÀI LIỆU THAM KHẢO .....	45

## **Phần thứ nhất**

### **MỞ ĐẦU**

Số học là một nội dung rất quan trọng nó giúp cho chúng ta hiểu cách biểu diễn và xử lý thông tin trong máy tính. Khi mới học lập trình học sinh bắt đầu bằng giải các bài toán số học cơ bản như: kiểm tra tính nguyên tố của một số nguyên dương, tìm ước chung lớn nhất của 2 số nguyên dương, phân tích một số nguyên dương thành tích các thừa số nguyên tố, các bài toán chia hết, ... Có rất nhiều bài toán khó khi giải bằng các giải thuật trừu tượng khó hiểu đôi khi không hiệu quả. Ta có thể phân tích bài toán và giải nó bằng các thuật toán số học sẽ hiệu quả hơn.

Để học sinh có thể lập trình giải các bài toán số học tốt cần trang bị cho các em các kiến thức cơ bản về số học:

- Biểu diễn thông tin trên máy tính bằng hệ nhị phân và hệ hexa
- Các khái niệm chia hết và đồng dư
- Một số thuật toán đếm cơ bản
- Các thuật toán về số nguyên tố
- Một số dãy số có tính chất đặc biệt
- Lý thuyết về tập hợp
- Một số nguyên lý số học cơ bản.

Tuy nhiên có nhiều bài toán có thuật toán khá đơn giản nhưng đòi hỏi phải biểu diễn các số có giá trị rất lớn. Giá trị của nó có thể lên đến hàng trăm chữ số. Nếu biểu diễn nó bằng các kiểu dữ liệu số thông thường như int, long, long long để áp dụng các phép toán số học gặp rất nhiều khó khăn, thậm chí không biểu diễn được. Trong ngôn ngữ lập trình có thể biểu diễn các số nguyên lớn bằng mảng hoặc xâu nhưng để thực hiện được các phép toán số học thông thường trên các kiểu dữ liệu này cần xây dựng các phép toán dưới dạng hàm bằng các thuật toán cụ thể.

Để phục vụ cho giảng dạy lớp chuyên tin chuyên đề số học và bồi dưỡng học sinh giỏi quốc gia. Chuyên đề này tôi hệ thống lại các phép toán xử lý số nguyên lớn bằng kiểu dữ liệu xâu ký tự trong sách giáo khoa chuyên tin quyển 1 và cài đặt một số bài toán về số học xử lý số lớn.

Rất mong được chia sẻ và nhận được sự đóng góp của quý thầy cô.

## Phần thứ hai

# NỘI DUNG CHUYÊN ĐỀ

## I. CÁC PHÉP TOÁN XỬ LÝ SỐ NGUYÊN LỚN

### I.1. Biểu diễn số nguyên lớn

Thông thường người ta sử dụng các cách biểu diễn số nguyên lớn sau:

\* **Xâu ký tự hoặc mảng xâu:** Đây là cách biểu diễn tự nhiên và đơn giản nhất, mỗi ký tự của một xâu tương ứng với một chữ số của số nguyên lớn tính từ trái qua phải.

\* **Mảng các số:** Sử dụng mảng lưu các chữ số (hoặc nhóm chữ số), và một biến ghi nhận số chữ số để thuận tiện trong quá trình xử lý.

\* **Danh sách liên kết các số:** Sử dụng danh sách liên kết các chữ số (hoặc một nhóm chữ số), cách làm này sẽ linh hoạt hơn trong việc sử dụng bộ nhớ.

### I.2. Các phép toán xử lý số nguyên lớn

#### I.2.1. Cộng 2 số nguyên lớn

##### a. Phân tích thuật toán

**Bước 1:** Chuẩn hóa hai xâu a, b để có độ dài bằng nhau. Nếu xâu nào có độ dài ngắn hơn thì thêm các '0' vào đầu xâu đó.

**Bước 2:** Duyệt từ cuối hai xâu về đầu xâu:

+ Tạo xâu kết quả c=a;

+ Tách từng phần tử của hai xâu chuyển sang kiểu số

+ Tính tổng:

**$tổng = số\ 1 + số\ 2 + nhớ\ (ban\ đầu\ nhớ\ bằng\ 0);$**

**$nhớ = tổng / 10;$**

**$tổng = tổng \% 10;$**

+ Chuyển đổi giá trị **tổng** tính được sang ký tự rồi gán vào xâu kết quả.

+ Lưu ý cộng thêm giá trị nhớ lần cuối nếu **nhớ khác '0'**

## b. Chương trình

```
string Congxau(string a, string b)
{
    string c;
    long n1=a.length(), n2=b.length(), i, nho=0, Tong;
    if(n1>n2) b.insert(0, n1-n2, '0');
    if(n1<n2) a.insert(0, n2-n1, '0');
    c=a;
    for (i=a.length()-1; i>=0; i--)
    {
        Tong=(a[i]-48)+(b[i]-48)+nho;
        nho=Tong/10;
        Tong=Tong%10;
        c[i]=char(Tong+48);
    }
    if(nho>0) c=char(nho+48)+c;
    return c;
}
```

### I.2.2. Trừ 2 số nguyên lớn (Trừ số lớn cho số bé)

#### a. Phân tích thuật toán

**Bước 1:** Chuẩn hóa hai xâu a, b để có độ dài bằng nhau. Nếu xâu nào có độ dài ngắn hơn thì thêm các '0' vào đầu xâu đó.

**Bước 2:** Duyệt từ cuối hai xâu về đầu xâu:

- + Tạo xâu kết quả c=a;
- + Tách từng phần tử của hai xâu chuyển sang kiểu số
- + Tính hiệu:

***hiệu = số 1 – số 2 - mượn (ban đầu mượn bằng 0);***

***Nếu hiệu<0 thì { hiệu=hiệu+10; mượn=1;}***

***Nếu hiệu>0 thì mượn =0;***

- + Chuyển đổi giá trị **hiệu** tính được sang ký tự rồi gán vào xâu kết quả.

- Xử lý xâu kết quả nếu xâu có độ dài lớn hơn 1 mà phần tử đầu tiên của mảng xâu là '0'.

#### b. Chương trình

```
string Truxau(string a, string b)
{
    string c="";
    long n1=a.length(), n2=b.length(), i, Muon=0, Hieu;
    if(n1>n2) b.insert(0, n1-n2, '0');
    for(i=a.length()-1; i>=0; i--)
    {
        Hieu=(a[i]-48)-(b[i]-48)-Muon;
        if(Hieu<0){Hieu+=10; Muon=1;}else Muon=0;
        c=char(Hieu+48)+c;
    }
    while(c.length()>1&& c[0]=='0') c.erase(0, 1);
    return c;
}
```

### I.2.3. Nhân một số nguyên lớn với một nguyên số nhỏ

#### a. Phân tích thuật toán

**Bước 1:** Duyệt từ cuối xâu số lớn về đầu xâu

**Bước 2:**

+ Tách từng phần tử của xâu chuyển sang kiểu số và tính tích

**$tích = số\ nhỏ * tg + nhớ$  (tg là số được tách từ xâu số lớn);**

**$nhớ = tích / 10;$**

**$Tích = tích \% 10;$**

+ Chuyển đổi giá trị **hiệu** tính được sang ký tự rồi gán vào xâu kết quả.

+ Lưu ý cộng thêm giá trị nhớ lần cuối nếu **nhớ khác '0'**

## b. Chương trình

```
string Nhanlso(string a, int k)
{
    string b;
    long i,Nho=0,Tich;
    for(i=a.length()-1;i>=0;i--)
    {
        Tich=Nho+(a[i]-48)*k;
        Nho=Tich/10;
        Tich=Tich%10;
        b=b+char(Tich+48);
    }
    if(Nho!=0) b=char(Nho+48)+b;
    while(b.length()>1&&b[0]=='0') b.erase(0,1);
    return b;
}
```

### I.2.4. Nhân 2 số nguyên lớn

#### a. Phân tích thuật toán

- Duyệt từ cuối xâu a về đầu xâu.
- Tách từng phần tử của xâu a nhân với xâu b (Thuật toán nhân với số nhỏ)
- Cộng liên tiếp các kết quả thu được (lưu ý trước khi cộng 2 xâu thêm ký tự “0” vào sau xâu thứ 2)
- Xử lý các ký tự ‘0’ trước xâu xau khi cộng.

## b. Chương trình

```
string Nhanxau(string a, string b)
{
    string x,Tg1="0",Tg2,c;
    long i,j=0;
    for (i=b.length()-1;i>=0;i--)
    {
        Tg2=Nhan1so(a, (b[i]-48));
        Tg2.insert(Tg2.length(),j,'0');
        j++;
        c=Congxau(Tg1,Tg2);
        Tg1=c;
    }
    return c;
}
```

### I.2.5. Chia số nguyên lớn cho số nguyên nhỏ

**Bước 1:** Duyệt từ đầu xâu số nguyên lớn

**Bước 2:**

+ Tách từng phần tử của xâu đem chia cho số nguyên nhỏ

*$chia = số + dư * 10$  (dư ban đầu bằng 0);*

*$thương = chia / số\ nhỏ$ ;*

*$dư = chia \% 10$ ;*

+ Cộng liên tiếp các thương được phần nguyên

+ Lưu lại giá trị dư cuối cùng được phần dư

+ Lưu ý: xóa các “0” ở đầu mảng xâu kết quả.



## b. Chương trình

```
void chia_so(char a[], long b, char div[], char mod[])
{
    long i, n = strlen(a), du = 0, so, chia, thuong;
    char tg[10], luu[100000] = "";
    for (i = 0; i < n; i++)
    {
        strncpy(tg, a + i, 1); tg[1] = '\0'; so = atoi(tg);
        chia = du * 10 + so; du = chia % b; thuong = chia / b;
        itoa(thuong, luu, 10);
        strcat(div, luu);
    }
    itoa(du, luu, 10); strcpy(mod, luu);
    i = 0;
    while (i < strlen(div) - 1 && div[i] == '0') i++;
    strcpy(luu, "");
    strncpy(luu, div + i, strlen(div) - i);
    luu[strlen(div) - i] = '\0';
    strcpy(div, luu);
}
```

### I.2.6. Chia hai số nguyên lớn

#### a. Phân tích thuật toán

- Lấy số ký tự của xâu a bằng số ký tự của xâu b lưu vào xâu chia
- Chừng nào số xâu chia còn lớn hơn xâu b thì tiến hành trừ liên tiếp xâu chia cho xâu b, ghi lại số lần trừ có thể là thương tìm được
- Hạ từng ký tự của xâu a xuống xâu chia
- Lưu ý: xử lý các ký tự “0” vô nghĩa của xâu a và xâu b.

## **b. Chương trình**

```
string Chiaxau(string a,string b,string &mod)
{
    string div="",Tg="",Tg1="";
    char luu[3];
    long i,n1,n2,dem;
    n1=a.length();n2=b.length();
    if((n1<n2)|| (n1==n2)&&(a<b)){mod=a;return "0";}
    else
    {
        Tg=a.substr(0,n2);
        for(i=n2-1;i<a.length();i++)
        {
            dem=0;Tg=Tg+a[i];
            while(Tg.length()>1&&Tg[0]=='0')
                Tg.erase(0,1);
            while((Tg.length()>n2)|| (Tg.length()==n2)&&(Tg>=b))
            {dem++;Tg=Truxau(Tg,b);}
            itoa(dem,luu,10);
            div=div+luu;
        }
        mod=Tg;while((mod.length()>1)&&(mod[0]=='0'))
            mod.erase(0,1);
    }
    return div;
}
```

## II. BÀI TẬP VẬN DỤNG

### II.1. Tách N thành tổng các số fibonacci (bài 2.11 sgk chuyên tin quyển 1)

#### a. Bài toán

Viết chương trình nhập vào số nguyên dương  $N$  ( $N \leq 10^{100}$ ), hãy tách  $N$  thành tổng các số Fibonacci đôi một khác nhau. Ví dụ, với  $N=19=1+5+13$ .

\* *Input*: đọc từ file văn bản TACHFIB.INP gồm 1 số nguyên dương  $N$

\* *Output*: ghi ra file văn bản TACHFIB.OUT gồm nhiều dòng, mỗi dòng ghi 1 số fibonacci tách được từ  $N$

\* *Example*

TACHFIB.INP	TACHFIB.OUT
16	13 3

#### b. Thuật toán

*Bước 1.* Tạo ra dãy số FIB gồm các số không vượt quá  $N$

*Bước 2.* Lặp lại bước 2.1 và 2.2 sau cho đến khi  $N=0$

2.1. Tìm số FIB[i] lớn nhất nhỏ hơn  $N$

2.2. Tách  $N$  theo FIB[i]

#### c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
string N, S[500];
string X1, X2, X3;
long count_;
int lS;
int smaller(string X, string Y)
{
    int l, lx = X.length(), ly = Y.length();
    if(lx>ly) l=lx; else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    if (X>Y) return 0;
```

```
        return 1;
    }
    string add(string X, string Y)
    {
        string Z="";
        int l,lx = X.length(),ly = Y.length();
        if(lx>ly) l=lx; else l=ly;
        int carry=0, x, y, sum;
        while (X.length() < l) X = '0' + X;
        while (Y.length() < l) Y = '0' + Y;
        for (int i= l-1; i>=0; i--)
        {
            x = (int) X[i] - 48;
            y = (int) Y[i] - 48;
            sum = x + y + carry;
            carry = sum/10;
            Z= (char) (sum%10 + 48) + Z;
        }
        if (carry>0) Z = '1' + Z;
        return Z;
    }
    void creat_F()
    {
        X1="0"; S[0]="0";
        X2="1"; S[1]="1";
        X3= add(X1,X2); lS=1;
        while (smaller(X3,N))
        {lS++; S[lS]=X3;X1=X2;X2=X3;X3= add(X1,X2);}
```

```
}  
  
int bigger2(string X, string Y)  
{  
    int l,lx = X.length(),ly = Y.length();  
    if(lx>ly) l=lx; else l=ly;  
    while (X.length() < l) X = '0' + X;  
    while (Y.length() < l) Y = '0' + Y;  
    if (X>Y) return 1;  
    return 0;  
}  
  
string sub (string X, string Y)  
{  
    string Z="";  
    int l,lx = X.length(),ly = Y.length();  
    if(lx>ly) l=lx; else l=ly;  
    int borrow=0, x, y, sum;  
    while (X.length() < l) X = '0' + X;  
    while (Y.length() < l) Y = '0' + Y;  
    for (int i= l-1; i>=0; i--)  
    {  
        x =(int)  X[i];  y  =  (int)  Y[i];  sum=x-y-  
borrow;  
        if(sum<0){sum  +=10;borrow=1;}else  borrow  =  
0;  
        Z = (char) (sum%10 + 48) + Z;  
    }  
    while((Z.length()>1)&&(Z[0]=='0')) Z.erase(0,1);  
    return Z;  
}
```

```
}  
  
void process()  
{  
    string X = S[lS];  
    while (N!="0")  
    {  
        cout<<X<< endl;  
        N = sub(N, X);  
        if (N=="0") break;  
        while (biger2(S[lS],N)) lS--;  
        X= S[lS];  
    }  
}  
  
int main()  
{  
    freopen("TACHFIB.INP","r",stdin);  
    freopen("TACHFIB.OUT","w",stdout);  
    getline(cin, N);  
    creat_F();  
    process();  
    return 0;  
}
```

**d. Test**

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

## II.2. Chuyển cơ số (bài 2.13 sgk chuyên tin quyển 1)

### a. Bài toán

Cho  $s$  là một xâu mô tả số nguyên không âm ở hệ cơ số  $a$ , hãy chuyển số đó sang hệ cơ số  $b$  ( $1 < a, b \leq 16$ , độ dài xâu  $s$  không vượt quá 50)

\* *Input*: đọc từ file văn bản DOICS.INP gồm:

- Dòng 1 ghi xâu  $s$
- Dòng 2 ghi 2 số  $a$  và  $b$  ( $a$  là cơ số của số biểu diễn bởi xâu  $s$ ,  $b$  là cơ số của số đổi từ cơ số  $a$  sang)

\* *Output*: ghi ra file văn bản DOICS.OUT một số duy nhất của hệ cơ số  $b$

\* *Example*

DOICS.INP	DOICS.OUT
100	1100100
10 2	

### b. Thuật toán

- Sử dụng các phép toán trên số lớn
- Chuyển số đếm ở hệ cơ số  $a$  sang hệ cơ số 10
- Chuyển số đếm ở hệ cơ số 10 sang hệ cơ số  $b$

### c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
string A, B;
long count_;
int lS, csa, csb;
string H = "0123456789ABCDEF";
string add(string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int carry=0,x,y,sum;
    if(lx>ly) l=lx; else l=ly;
```

```
while (X.length() < l) X = '0' + X;
while (Y.length() < l) Y = '0' + Y;
for (int i= l-1; i>=0; i--)
{
    x = (int) X[i] - 48;
    y = (int) Y[i] - 48;
    sum = x + y + carry;
    carry = sum/10;
    Z = (char) (sum%10 + 48) + Z;
}
if (carry>0) Z = '1' + Z;
return Z;
}
string multiply_1(string X, long b)
{
    long carry=0,s,lx = X.length();
    string Z="";
    for (int i= lx-1; i>=0; i--)
    {
        s = ((int) X[i] - 48 ) * b + carry;
        carry = s / 10;
        Z = (char) (s%10+48) + Z;
    }
    while (carry)
    {Z= (char) (carry%10 + 48) + Z;carry /= 10;}
    return Z;
}
string div_1(string X, long b)
```



```
{
    int s=0, hold=0, lx=X.length();
    string Z="";
    for (long i=0; i<lx; i++)
    {
        hold = hold*10 + (int)(X[i]) -48;
        s = hold/b; hold %= b; Z=Z+(char)(s + 48);
    }
    while((Z.length()>1)&&(Z[0]=='0')) Z.erase(0,1);
    return Z;
}

long mod_1(string X, long b)
{
    int hold=0, lx=X.length();
    for (long i=0; i<lx; i++)
        hold = ((int) X[i] -48 + hold * 10 ) % b;
    return hold;
}

string cs1_to_cs10(string s)
{
    string s2 = "";
    for (int i=0; i<s.length(); i++)
    {
        s2 = multiply_1(s2, csa); int pos=0;
        for (int p=0; p<16; p++)
            if (s[i]==H[p]) {pos = p;break;}
        string s3="";
        while(pos){s3=(char(pos%10+48)+s3;pos/= 10;}
```

```
        s2 = add (s2, s3);
    }
    return s2;
}
string cs10_to_csb(string s)
{
    string s2 = "";
    while (s != "0")
    {int      i=mod_1(s,      csb);s2=H[i]+s2;s=div_1(s,
csb);}
    return s2;
}
int main()
{
    freopen("DOICS.INP", "r", stdin);
    freopen("DOICS.OUT", "w", stdout);
    getline(cin, A);
    cin >> csa >> csb;
    cout<< cs10_to_csb(cs1_to_cs10(A));
    return 0;
}
```

**d. Test**

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

### II.3. Nguồn của một số (bài 2.18 sgk chuyên tin quyển 1)

#### a. Bài toán

Giả thiết N là số nguyên dương. Số nguyên M là tổng của N và các chữ số của nó. N được gọi là nguồn của M.

Ví dụ,  $N=245$ , khi đó  $M=245+2+4+5=256$ . Như vậy, nguồn của 256 là 245.

Có những số không có nguồn và có số lại có nhiều nguồn. Ví dụ, số 216 có 2 nguồn là 198 và 207.

Cho số nguyên M (M có không quá 100 chữ số) hãy tìm nguồn nhỏ nhất của nó. Nếu M không có nguồn thì đưa ra số 0.

\* *Input*: đọc từ file văn bản NGUON.INP gồm một số nguyên dương M

\* *Output*: ghi ra file văn bản NGUON.OUT một số duy nhất là nguồn nhỏ nhất của M.

\* *Example*

NGUON.INP	NGUON.OUT
216	198

#### b. Thuật toán

Theo định nghĩa, ta có  $M=N+S(N)$ , trong đó  $S(N)$  là tổng các chữ số của N. Nhận thấy  $S(N)<9 \times$  số chữ số của M, do đó ta thử  $S(N)$  để tìm N.

#### c. Chương trình

```
#include<bits/stdc++.h>
using namespace std;
string M,N,S,Nguon="0",Sum;
string Congxau(string a, string b)
{
    string c;
    long n1=a.length(),n2=b.length(),i,nho=0,Tong;
    if(n1>n2) b.insert(0,n1-n2,'0');
    if(n1<n2) a.insert(0,n2-n1,'0');
    c=a;
    for(i=a.length()-1;i>=0;i--)
    {
```

```
Tong=(a[i]-48)+(b[i]-48)+nho;
nho=Tong/10;
Tong=Tong%10;
c[i]=char(Tong+48);
}
if(nho>0)c=char(nho+48)+c;
return c;
}
string Truxau(string a, string b)
{
    string c="";
    long n1=a.length(),n2=b.length(),i,Muon=0,Hieu;
    if(n1>n2) b.insert(0,n1-n2,'0');
    for(i=a.length()-1;i>=0;i--)
    {
        Hieu=(a[i]-48)-(b[i]-48)-Muon;
        if(Hieu<0){Hieu+=10;Muon=1;}else Muon=0;
        c=char(Hieu+48)+c;
    }
    while(c.length()>1&&c[0]=='0') c.erase(0,1);
    return c;
}
string Tong(string s)
{
    string T="0",Tg;
    long n=s.length();
    for(int i=0;i<n;i++){Tg=s[i];T=Congxau(T,Tg);}
    return T;
}
```

```
string So_To_xau(long k)
{
    string Tg="";
    while(k!=0)
    {Tg=char(k%10+48)+Tg;k=k/10;}
    return Tg;
}
bool sosanh(string a, string b){
    long n1=a.length(),n2=b.length();
    if(n1<n2) return 1;
    if(n1==n2&&a<=b) return 1;
    return 0;
}
int main(){
    freopen("NGUON.INP","r",stdin);
    freopen("NGUON.OUT","w",stdout);
    cin>>M;
    long so=9*M.length()+1;
    N=Truxau(M,So_To_xau(so));
    while(sosanh(N,M)){
        S=Tong(N);
        Sum=Congxau(S,N);
        if(M==Sum){Nguon=N; break;}
        N=Congxau(N,"1");
    }
    cout<<Nguon;
}
```

**d. Test**

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

## II.4. Số các ước và tổng ước của N! (bài 2.19 sgk chuyên tin quyển 1)

### a. Bài toán

Tính số ước và tổng các ước của N! ( $N \leq 100$ )

\* *Input*: đọc từ file văn bản TONGUOC.INP gồm 1 số nguyên dương N

\* *Output*: đưa ra file văn bản TONGUOC.OUT gồm:

- Dòng 1 ghi số các ước của N!
- Dòng 2 ghi tổng các ước của N!

\* *Example*

TONGUOC.INP	TONGUOC.OUT
5	16
	360

### b. Thuật toán

Nếu phân tích số nguyên N dưới dạng tích các thừa số nguyên tố là  $N = a_1^{i_1} \cdot a_2^{i_2} \cdot \dots \cdot a_k^{i_k}$  thì số các ước số của N là  $(i_1+1)(i_2+1)\dots(i_k+1)$  và

tổng các ước của N là  $[(a_1^{i_1+1}-1)/(a_1-1)] \cdot [(a_2^{i_2+1}-1)/(a_2-1)] \dots [(a_k^{i_k+1}-1)/(a_k-1)]$

### c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
#define maxN 101
long long N, A[maxN];
string add(string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int carry=0,x,y,sum;
    if(lx>ly) l=lx; else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    Z=X;

```

```
    for (int i= l-1; i>=0; i--)
    {
        x = (int) X[i] - 48;
        y = (int) Y[i] - 48;
        sum = x + y + carry;
        carry = sum/10;
        Z[i] = (char) (sum%10 + 48);
    }
    if (carry>0) Z = '1' + Z;
return Z;
}
string sub (string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    if(lx>ly) l=lx; else l=ly;
    int borrow=0,x,y,sum;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    Z=X;
    for (int i= l-1; i>=0; i--)
    {
        x = (int) X[i]; y = (int) Y[i];
        sum = x - y - borrow;
        if(sum<0{sum+=10;borrow= 1;} else borrow =
0;
        Z[i] = (char) (sum%10 + 48);
    }
    while((Z.length()>1)&&(Z[0]=='0'))      Z.erase(0,
1);
```

```
        return Z;
    }
    string multiply_1(string X, long b)
    {
        long carry=0, s, lx = X.length();
        string Z=X;
        for (int i= lx-1; i>=0; i--)
        {
            s = ((int) X[i] - 48 ) * b + carry;
            carry = s / 10;
            Z[i] = (char) (s%10+48);
        }
        while(carry){Z=(char(carry%10+48)+Z;carry /=
10;}
        while((Z.length()>1)&&(Z[0]=='0'))        Z.erase(0,
1);
        return Z;
    }
    string multiply(string X, string Y)
    {
        long m, lx = X.length(), ly = Y.length() ;
        string Z="", tmp="";
        m = -1;
        for (int i=lx-1; i>=0; i--)
        {
            m++;
            tmp = multiply_1(Y, (int) X[i] - 48);
            for (int j=0; j<m; j++) tmp = tmp + "0";
            Z = add(tmp, Z);
        }
    }
```



```
        return Z;
    }
    string chuyen(long x)
    {
        string Z="";
        while (x>0) {Z = char(x%10 + 48) + Z; x /= 10;}
        return Z;
    }
    void analyse(long i)
    {
        long ts = 2;
        while (i != 1)
            {while (i%ts==0) {A[ts]++;i /= ts;}ts++;}
    }
    void process()
    {for (long i=2; i<=N; i++)analyse(i);}
    void calculatel()
    {
        string res1 = "";res1 = res1 + "1";
        for (long i=2; i<=maxN; i++)
            if (A[i]>0)
                {
                    string tmp = chuyen(A[i]);
                    tmp = add(tmp, "1");
                    res1 = multiply(res1, tmp);
                }
        cout << res1 << endl;
    }
```

```
void calculate3()
{
    string res3 = ""; res3 = res3 + "1";
    for (long i=2; i<=maxN; i++)
        if (A[i]>0)
        {
            string tmp0= ""; tmp0=tmp0 + "0";
            string tmp = chuyen(i);
            string tmp1= ""; tmp1=tmp1 + "1";
            for (int j=1; j<=A[i]; j++)
            {
                tmp1= multiply(tmp1, tmp);
                tmp0 = add(tmp0,tmp1);
            }
            tmp0 = add(tmp0,"1");
            res3 = multiply(res3, tmp0);
        }
    cout << res3 << endl;
}

int main()
{
    freopen("TONGUOC.INP","r",stdin);
    freopen("TONGUOC.OUT","w",stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cin>>N;
    process();
    calculatel();
    calculate3();
    return 0;
}
```

#### d. Test

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

### II.5. Cân đĩa (Bài 2.20 sgk chuyên tin quyển 1)

#### a. Bài toán

Cho một chiếc cân hai đĩa và các quả cân có khối lượng  $3^0, 3^1, 3^2, \dots$

Hãy chọn các quả cân để có thể cân được vật có khối lượng  $N$  ( $N < 10^{100}$ )

*Ví dụ:* cân cân vật có khối lượng  $N=11$  ta cần sử dụng các quả cân sau:

- Cân bên trái: quả cân  $3^1$  và  $3^2$
- Cân bên phải: quả cân  $3^0$  và vật  $N=11$

\* *Input:* đọc từ file văn bản CANDIA.INP gồm một số  $N$

\* *Output:* ghi ra file văn bản CANDIA.OUT gồm 2 dòng:

- Dòng 1: ghi các giá trị của quả cân ở đĩa bên trái
- Dòng 2: ghi giá trị các quả cân ở đĩa bên phải (bên có vật)

\* *Example*

CANDIA.INP	CANDIA.OUT
11	3 9 1

#### b. Thuật toán

Biểu diễn  $N$  ở hệ cơ số 3, mỗi bit ứng với một loại quả cân: bit 0 ứng với quả  $3^0$ , bit 1 ứng với quả  $3^1, \dots$ . Do mỗi loại quả cân chỉ có 1 quả nên trong biểu diễn  $N$  ở hệ cơ số 3 nếu gặp bit nào có giá trị 2 thì phải thêm vào bit đó giá trị 1 (nghĩa là thêm vào đĩa có vật một quả cân tương ứng với bit này), sau đó bit này phải dồn lên thành giá trị 1 thêm vào bit cao hơn bên trái nó. Quá trình dồn bit tiếp tục về bên trái nếu khi dồn vẫn gặp giá trị 3.

#### c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
string N, B;
long long count_;
long long lS, csb=3, csa=3;
```

```
string H = "0123456789ABCDEF";

string div_1(string X, long long b)
{
    long long s=0, hold=0, lx=X.length();
    string Z="";
    for (long i=0; i<lx; i++)
    {
        hold = hold*10 + (int)(X[i]) -48;
        s = hold/b; hold %= b;
        Z = Z + (char) (s + 48);
    }
    while((Z.length()>1)&& (Z[0]=='0'))
        Z.erase(0,1);
    return Z;
}

long mod_1(string X, long long b)
{
    long long hold=0, lx=X.length();
    for (long i=0; i<lx; i++)
        hold = ((int) X[i] -48 + hold * 10 ) % b;
    return hold;
}

string cs10_to_csb(string s)
{
    string s2 = "";
    while (s != "0")
    {
        int i = mod_1(s, csb);
        s2 = H[i] + s2; s = div_1(s, csb);
    }
}
```

```
return s2;
}
string add(string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int carry=0;
    long long x, y, sum;
    if(lx>ly) l=lx; else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--)
    {
        x = (int) X[i] - 48;
        y = (int) Y[i] - 48;
        sum = x + y + carry;
        carry = sum/10;
        Z = (char) (sum%10 + 48) + Z;
    }
    if (carry>0) Z = '1' + Z;
    return Z;
}
string multiply_1(string X, long long b)
{
    long long carry=0, s, lx = X.length();
    string Z="";
    for (int i= lx-1; i>=0; i--)
    {
```

```
        s = ((int) X[i] - 48 ) * b + carry;
        carry = s / 10;
        Z = (char) (s%10+48) + Z;
    }
    while (carry)
    {Z= (char) (carry%10 + 48) + Z;carry /= 10;}
    return Z;
}
string sub (string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int borrow=0;
    long long x, y, sum;
    if(lx>ly) l=lx; else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--)
    {
        x=(int)X[i];y =(int)Y[i];sum=x-y-borrow;
        if(sum <0){sum+=10;borrow=1;}else borrow =
0;
        Z = (char) (sum%10 + 48) + Z;
    }
    while((Z.length()>1)&& (Z[0]=='0'))
Z.erase(0,1);
    return Z;
}
```

```
string cs1_to_cs10(string s)
{
    string s2 = "";
    for (int i=0; i<s.length(); i++)
    {
        s2 = multiply_1(s2, csa);
        int pos=0;
        for (int p=0; p<16; p++)
            if (s[i]==H[p]) {pos = p;break;}
        string s3="";
        while(pos){s3=(char) (pos%10+48)+s3;pos/=0;}
        s2 = add (s2, s3);
    }
    return s2;
}

void process2(){
    string res = "", tmp="";
    res = cs10_to_csb(N);
    string luu_res = res;
    res = "0" + res;
    long long l = res.length();
    long long nho=0;
    for (int i=l-1; i>=0; i--){
        long long c = (int) (res[i]) -48 + nho;
        if (c>1) {res[i] = char(48);nho = 1;}
        if (c==1){res[i]=char(49); nho = 0; }
    }
    string quacan = "1";
```

```
for (int i=res.length()-1; i>=0; i--){
    if ((int) res[i] - 48 >0) cout<<quacan<<" ";
    quacan =multiply_1(quacan, 3);
}
cout<<endl;
string kq1 = cs1_to_cs10(luu_res);
string kq2 = cs1_to_cs10(res);
string res0 = sub(kq2, kq1);
res = cs10_to_csb(res0);
quacan = "1";
bool them = false;
for (int i=res.length()-1; i>=0; i--){
    if((int)res[i]-48>0){cout<<quacan<<" ";them= true;}
    quacan =multiply_1(quacan, 3);}
    if (them==false) cout << 0;
}
int main(){
    freopen("CANDIA.INP","r",stdin);
    freopen("CANDIA.OUT","w",stdout);
    getline(cin,N);
    process2();
    return 0;
}
```

#### d. Test

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

## II.6. Hoán vị xâu (bài 2.22 sgk chuyên tin quyển 1)

### a. Bài toán

Cho xâu s chỉ gồm kí tự từ 'a' đến 'z' (độ dài xâu s không vượt quá 100), hãy đếm số hoán vị khác nhau của xâu đó. *Ví dụ:* s='aba', ta có 3 hoán vị 'aab',



'aba', 'baa'

\* *Input*: đọc từ file văn bản HVXAU.INP gồm một chuỗi s

\* *Output*: ghi ra file văn bản HVXAU.OUT gồm một số duy nhất là số hoán vị khác nhau của s

\* *Example*

HVXAU.INP	HVXAU.OUT
abac	12

## b. Thuật toán

Số hoán vị của chuỗi s bằng  $\text{length}(s) ! / ((\text{số kí tự a})! \dots (\text{số kí tự z})!)$

## c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
string s="";
int d[27];
string multiply_1(string X, long b)
{
    long carry=0, s, lx = X.length();
    string Z="";
    for (int i= lx-1; i>=0; i--)
    {
        s = ((int) X[i] - 48 ) * b + carry;
        carry = s / 10;
        Z = (char) (s%10+48) + Z;
    }
    while (carry)
        {Z= (char) (carry%10 + 48) + Z; carry /= 10;}
    return Z;
}
```

```
string div_1(string X, long b)
{
    int s=0, hold=0, lx=X.length();
    string Z="";
    for (long i=0; i<lx; i++){
        hold = hold*10 + (int)(X[i]) -48;
        s = hold/b; hold %= b;
        Z = Z + (char) (s + 48);
    }
    while((Z.length()>1)&&(Z[0]=='0')) Z.erase(0,1);
    return Z;
}

void calculate (string s)
{
    int ls = s.length();
    for (int i=0; i<ls; i++) d[(int) s[i] - 96]++;
    string res = "1";
    for (int i=1; i<=ls; i++) res= multiply_1(res,
i);
    for (int i=0 ; i<26 ; i++)
        if (d[i]>1)
            for(int j=2;j<=d[i];j++) res=div_1(res,
j);
    cout << res;
}

int main()
{
    freopen("HVXAU.INP", "r", stdin);
    freopen("HVXAU.OUT", "w", stdout);
```

```
getline(cin, s);  
calculate(s);  
return 0;  
}
```

**d. Test**

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

**II.7. Đánh số trang sách (bài 2.23 sgk chuyên tin quyển 1)**

**a. Bài toán**

Nam quyết định đánh số trang cho quyển sách của anh ta từ 1 đến N. Hãy tính toán số lượng chữ số 0 cần dùng, số lượng chữ số 1 cần dùng,..., số lượng chữ số 9 cần dùng.

\* *Input*: đọc từ file văn bản DIGITS.INP gồm 1 dòng duy nhất chứa một số N ( $N \leq 10^{100}$ ).

\* *Output*: ghi ra file văn bản DIGITS.OUT có dạng gồm 10 dòng

- Dòng thứ nhất là số lượng chữ số 0 cần dùng,
- Dòng thứ hai là số lượng chữ số 1 cần dùng,...
- Dòng thứ 10 là số lượng chữ số 9 cần dùng.

\* *Example*

DIGITS.INP	DIGITS.OUT
10	1
	2
	1
	1
	1
	1
	1
	1
	1
	1

**b. Thuật toán**

Có 10 số nguyên không âm 1 chữ số. Vậy tổng số các chữ số cần dùng là  $10 \times 1$ . Mỗi chữ số xuất hiện như nhau. Do đó xét mọi số nguyên một chữ số thì mỗi chữ số được đếm 1 lần, ngoại trừ số 0 không được đếm vì đề bài xét các số từ số 1 (đến N).

Có 100 số nguyên không âm có một hoặc hai chữ số là: 00, 01, ...99. Tổng các chữ số của chúng là  $100 \times 2 = 200$ , Các chữ số xuất hiện như nhau. Do đó khi xét trong mọi số nguyên có một hoặc hai chữ số thì mỗi chữ số được dùng 20 (lần)  $= 2 \times 10^1$ , ngoại trừ số 0 chỉ được đếm 9 (lần)  $= 2 \times 10^1 - 11$  (11 số 0 trong các số sau không được tính 00, 01, ..., 09).

Có 1000 số nguyên không âm có một hoặc hai hoặc ba chữ số là: 000, 001, ...999. Tổng các chữ số của chúng là  $1000 \times 3 = 3000$ , Các chữ số xuất hiện như nhau. Do đó khi xét trong mọi số nguyên có một hoặc hai hoặc 3 chữ số thì mỗi chữ số được dùng 300 (lần)  $= 3 \times 10^2$ , ngoại trừ số 0 chỉ được đếm 189 (lần)  $= 3 \times 10^2 - 111$  (111 số 0 trong các số sau không được tính 000, 001, ..., 009, 010, 011, 099).

Bây giờ xem xét số nguyên N có k chữ số.

Số số 0 vô nghĩa của các số từ 1 đến N (N có k chữ số) là số có dạng 11...11 (k số 1 liên nhau).

Giả sử chữ số đầu tiên là d (chữ số thuộc hàng cao nhất - chữ số bên trái nhất)

Gọi bài toán A là tìm số lượng mỗi loại chữ số khi dùng các số từ 1 đến N

Bài toán B là tìm số lượng mỗi loại chữ số khi dùng các số từ 1 đến  $N \% 10^{k-1}$  (nghĩa là N đã bỏ đi một chữ số hàng cao nhất của nó)

Các chữ số từ 0 đến d-1 đứng đầu mỗi khối gồm các số có k-1 chữ số (từ 00..0 đến 99..9), vậy có d khối mỗi khối này có chiều cao là  $10^{k-1}$ , chiều rộng là k-1, do đó nó gồm  $(k-1) \cdot 10^{k-1}$  chữ số. Các loại chữ số trong mỗi khối xuất hiện như nhau nên số lượng mỗi loại chữ số là  $(k-1) \cdot 10^{k-1} : 10 = (k-1) \cdot 10^{k-2}$  Vậy mỗi loại chữ số trong bài A hơn trong bài B là  $d \cdot 10^{k-2} (k-1)$ . Ngoài ra các chữ số từ 0 đến d-1 đứng ở hàng cao nhất của bài toán A còn đứng đầu các khối của bài toán B nên chúng được tăng thêm một lượng bằng chiều cao  $10^{k-1}$ . Riêng chữ số d trong A chỉ hơn trong B là chiều cao của các số trong bài toán B cộng thêm 1 (bằng  $N \% 10^{k-1} + 1$ )

Tương tự xét lần lượt từng chữ số ở hàng cao nhất còn lại của số N sau mỗi lần loại bỏ dần các chữ số này.

Cuối cùng, với số lượng chữ số 0 đã thu được cần trừ đi số lượng các số 0 vô nghĩa.

### c. Chương trình

```
#include <bits/stdc++.h>
using namespace std;
string N;
```

```
int d, k;
string ones = "";
string tens = "", tens1="";
string res[10];

void read_input()
{
    getline(cin, N);
    k = N.length(); // số chữ số của N
    for (int i=0; i<k ; i++) ones += "1";
    tens = "1";
    for (int i=0; i<k-1; i++) tens = tens + "0";
}

string add(string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int carry=0, x, y, sum;
    if(lx>ly) l=lx; else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--)
    {
        x = (int) X[i] - 48;
        y = (int) Y[i] - 48;
        sum = x + y + carry;
        carry = sum/10;
        Z = (char) (sum%10 + 48) + Z;
    }
}
```

```
        if (carry>0) Z = '1' + Z;
return Z;
}
string sub (string X, string Y){
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    if(lx>ly) l=lx; else l=ly;
    int borrow=0, x, y, sum;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--)
        {x = (int) X[i];y =(int) Y[i];sum = x - y -
borrow;
        if (sum <0){sum+=10;borrow = 1;} else borrow =
0;
        Z = (char) (sum%10 + 48) + Z;}
    while((Z.length()>1)&& (Z[0]=='0'))
Z.erase(0,1);
    return Z;
}
string mul(string X, long b)
{
    long carry=0, s, lx = X.length();
    string Z="";
    for (int i= lx-1; i>=0; i--)
    {
        s=((int) X[i]-48) * b + carry;
        carry = s / 10;
        Z = (char) (s%10+48) + Z;
    }
    while(carry){Z=(char) (carry%10+48)+Z;carry/=10;}
```

```
        return Z;
    }
    int main()
    {
        freopen("DIGITS.INP", "r", stdin);
        freopen("DIGITS.OUT", "w", stdout);
        read_input();
        while (tens != "")
        {
            d = (int) N[0] - 48;
            N = N.erase(0, 1);
            k--;
            tens1 = tens; // 10^k - 1
            tens1 = tens1.erase(tens1.length() - 1, 1);
            for (int i = 0; i < 10; i++)
                res[i] = add(res[i], mul(mul(tens1, k), d));
            for (int i = 0; i < d; i++) res[i] = add(res[i],
tens);
            res[d] = add(add(res[d], N), "1");
            tens = tens1;
        }
        res[0] = sub(res[0], ones);
        for (int i = 0; i < 10; i++) cout << res[i] << endl;
        return 0;
    }
```

#### d. Test

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

## II.8. Hái năm (bài 2.25 sgk chuyên tin quyển 1)

### a. Bài toán

Một cháu gái hàng ngày được mẹ giao nhiệm vụ đến thăm bà nội. Từ nhà mình đến nhà bà nội cô bé phải đi qua một khu rừng có rất nhiều loại nấm.

Trong số các loại nấm, có ba loại có thể ăn được. Cô bé đánh số ba loại nấm ăn được lần lượt là 1, 2 và 3. Là một người cháu hiếu thảo cho nên cô bé quyết định mỗi lần đến thăm bà, cô sẽ hái ít nhất hai loại nấm ăn được để nấu súp cho bà. Khu rừng mà cô bé đi qua được chia thành lưới ô vuông gồm  $m$  hàng và  $n$  cột. Các hàng của lưới được đánh số từ trên xuống dưới bắt đầu từ 1, còn các cột được đánh số từ trái sang phải, bắt đầu từ 1. Ô nằm giao của hàng  $i$  và cột  $j$  có tọa độ  $(i, j)$ . Trên mỗi ô vuông, trừ ô  $(1,1)$  và ô  $(m, n)$  các ô còn lại hoặc có nấm độc và cô bé không dám đi vào (đánh dấu là -1), hoặc là có đúng một loại nấm có thể ăn được (đánh dấu bằng số hiệu của loại nấm đó). Khi cô bé đi vào một ô vuông có nấm ăn được thì cô bé sẽ hái loại nấm mọc trên ô đó. Xuất phát từ ô  $(1,1)$ , để đến được nhà bà nội ở ô  $(m, n)$  một cách nhanh nhất cô bé luôn đi theo hướng sang phải hoặc xuống dưới.

Việc đi thăm bà và hái nấm trong rừng sâu gặp nguy hiểm bởi có một con chó sói luôn theo dõi và muốn ăn thịt cô bé. Để phòng tránh chó sói theo dõi và ăn thịt, cô bé quyết định mỗi ngày sẽ đi theo một con đường khác nhau (hai con đường khác nhau nếu chúng khác nhau ở ít nhất một ô).

*Yêu cầu:* Cho bảng  $m \times n$  ô vuông mô tả trạng thái khu rừng. Hãy tính số con đường khác nhau để cô bé đến thăm bà nội theo cách chọn đường đi đã nêu ở trên.

\* *Input:* đọc từ file văn bản MUSHROOM.INP gồm:

- Dòng đầu chứa 2 số  $m, n$  ( $1 < m, n < 101$ ),

-  $m$  dòng tiếp theo, mỗi dòng chứa  $n$  số nguyên cho biết thông tin về các ô của khu rừng. (riêng giá trị ở hai ô  $(1,1)$  và ô  $(m, n)$  luôn luôn bằng 0 các ô còn lại có giá trị bằng -1, hoặc 1, hoặc 2, hoặc 3).

Hai số liên tiếp trên một dòng cách nhau một dấu cách.

\* *Output:* Đưa ra file văn bản MUSHROOM.OUT chứa một dòng ghi một số nguyên là kết quả bài toán.

\* *Example*

MUSHROOM.INP	MUSHROOM.OUT
3 4	3
0 3 -1 2	
3 3 3 3	
3 1 3 0	



## b. Thuật toán

- Gọi W là số đường đi khác nhau từ ô (1;1) đến ô (M;N).
- Gọi A là số đường đi khác nhau từ ô (1;1) đến ô (M;N) chỉ đi qua ô nắm loại 1
- Gọi B là số đường đi khác nhau từ ô (1;1) đến ô (M;N) chỉ đi qua ô nắm loại 2
- Gọi C là số đường đi khác nhau từ ô (1;1) đến ô (M;N) chỉ đi qua ô nắm loại 3

Số đường đi khác nhau từ ô (1; 1) đến ô (M; N) đi qua ít nhất hai loại nắm là W-A-B-C

Đọc dữ liệu vào chuyển tới mảng kí tự A[M][N] trong đó A[i][j] = “0” với hai ô (1; 1) và (M; N); A[i][j] = “4” nếu là ô không được đi vào, A[i][j] = “1” hoặc “2” hoặc “3” tương ứng là ô có nắm loại 1, 2 hoặc 3.

Dùng mảng F[M][N] lưu số lượng đường đi khác nhau từ ô(1; 1) đến ô (M; N) theo một đặc điểm là loại W, hay A, hay B, hay C theo công thức:  $F[i; j] = F[i-1][j] + F[i][j-1]$  với điều kiện ô (i; j) thuộc đặc điểm đang xét.

Chú ý: ô có A[i][j]=“0” luôn đi vào được.

## c. Chương trình

```
#include <bits/stdc++.h>
#define maxMN 101
using namespace std;
long M, N;
string A[maxMN][maxMN];
string F[maxMN][maxMN];
void read_input()
{
    int x;
    cin >> M >> N;
    for (int i=1; i<=M; i++)
        for (int j=1; j<=N; j++)
        {
            cin >> x;
```

```
        if (x==1) A[i][j] = "1";
        else if (x==2) A[i][j] = "2";
            else if (x==3) A[i][j] = "3";
                else if (x==0) A[i][j] = "0";
                    else if (x==-1) A[i][j] =
"4";
            }
    }
string add(string X, string Y)
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int carry=0, x, y, sum;
    if(lx>ly) l=lx;else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--)
    {
        x = (int) X[i] - 48;
        y = (int) Y[i] - 48;
        sum = x + y + carry;
        carry = sum/10;
        Z = (char) (sum%10 + 48) + Z;
    }
    if (carry>0) Z = '1' + Z;
    return Z;
}
string sub (string X, string Y)
```

```
{
    string Z="";
    int l,lx=X.length(),ly=Y.length();
    int borrow=0, x, y, sum;
    if(lx>ly) l =lx;else l=ly;
    while (X.length() < l) X = '0' + X;
    while (Y.length() < l) Y = '0' + Y;
    for (int i= l-1; i>=0; i--){
        x = (int) X[i]; y = (int) Y[i];
        sum = x - y - borrow;
        if sum<0){sum+=10; borrow=1;} else borrow=0;
        Z = (char) (sum%10 + 48) + Z;
    }
    while((Z.length()>1)&&(Z[0]=='0'))      Z.erase(0,
1);
    return Z;
}
string cal(string mark){
    F[0][1] = "1"; F[1][0]="0";
    for (int i=1; i<=M; i++)
        for (int j=1; j<=N; j++)
            if ((int) mark.find(A[i][j]) >= 0)
                F[i][j] = add(F[i-1][j], F[i][j-1]);
    return F[M][N];
}
int main(){
    freopen("MUSHROOM.INP","r",stdin);
    freopen("MUSHROOM.OUT","w",stdout);
```

```
read_input();  
string res = sub(cal("0123"), cal("01"));  
res = sub(res, cal("02"));  
res = sub(res, cal("03"));  
cout<<res;  
return 0;  
}
```

**d. Test**

<http://www.mediafire.com/file/yk2e38w38wsrmqs/ExamsHV.rar>

**III. BÀI TẬP LUYỆN TẬP**

<http://vn.spoj.com/problems/CHUOIHAT/>

<http://vn.spoj.com/problems/LATGACH/>

<http://vn.spoj.com/problems/SPBINARY/>

<http://vn.spoj.com/problems/PBCDEM/>

<http://vn.spoj.com/problems/TREEPATH/>

### **Phần thứ ba**

## **KẾT LUẬN**

Chuyên đề này tôi mới trình bày, cài đặt các phép toán với số nguyên lớn và giải một số bài toán có thuật toán đơn giản có thể áp dụng trực tiếp các phép toán số học. Còn rất nhiều bài tập khó cần phải xử lý số nguyên lớn mà tôi chưa đề cập đến.

Thời gian viết chuyên đề có hạn, không tránh khỏi những sai sót, rất mong được sự góp ý kiến của các thầy cô.

*Cảm ơn các thầy cô đã đọc tài liệu!*

## **TÀI LIỆU THAM KHẢO**

Tài liệu sách giáo khoa chuyên tin – Hồ Sỹ Đàm chủ biên

Trang web: <http://www.vn.spoj.com>