# LẬP TRÌNH NÂNG CAO

*Một số Bài tập khó hơn để các em rèn luyện. Để làm tốt các bài tập này, các em cần phân tích đề thật kĩ, loại bỏ các ngữ cảnh rườm rà của bài toán, và phải biết kết hợp các kiến thức về ngôn ngữ lập trình lại với nhau.*

#### Taxi TAXI 158B



Sau giờ học n nhóm học sinh quyết định đến thăm Polycarpus để mừng sinh nhật cậu ấy. Biết rằng nhóm thứ i có *si (1 ≤ si ≤ 4)* bạn, và họ muốn đi đến nhà Polycarpus cùng với nhau. Họ quyết định đến đó bằng taxi. Mỗi chiếc xe có thể mang nhiều nhất là bốn hành khách. Hãy tính số lượng tối thiểu xe taxi bọn trẻ cần nếu tất cả các thành viên của cùng một nhóm phải đi cùng xe taxi (nhưng một taxi có thể có nhiều hơn một nhóm)?

**Input**

* Dòng đầu tiên chứa số nguyên n (1 ≤ n ≤ 105) - số lượng các nhóm học sinh.
* Dòng thứ hai chứa một chuỗi các số nguyên s1, s2, ..., sn (1 ≤ si ≤ 4). Các số nguyên được phân cách bởi một không gian, si là số lượng trẻ em trong nhóm thứ i.

**Output:** In số duy nhất - số lượng taxi tối thiểu cần thiết để chở tất cả.

**Ví dụ**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** |  |
| 5  1 2 4 3 3 | 4 | Có nhiều cách khác để sắp xếp các nhóm thành bốn xe ô tô. Đây là một cách   * nhóm thứ ba (có bốn người), * nhóm thứ tư (có ba người), * nhóm thứ năm (có ba người), * nhóm đầu tiên và nhóm thứ hai (bao gồm một người và hai người). |
| 8  2 3 4 4 2 1 3 1 | 5 |  |
| 10  2 3 4 4 2 3 3 3 1 2 | 8 |  |

##### Giải

d[i] là số lượng nhóm có i học sinh.

* Các nhóm 4 mỗi nhóm đi 1 xe
* Các nhóm 3 kết hợp với nhóm 1 nhiều nhất có thể
* Các nhóm 2 kết hợp với các nhóm 2 nhiều nhất có thể
* Các nhóm 2 kết hợp với các nhóm 1 nhiều nhất có thể
* Các nhóm 1 kết hợp với nhau.

#### Mã số MASO



Để tiện trong công tác quản lý, một công ty đã gắn mã số cho các nhân viên của mình. Mã số của mỗi nhân viên bao gồm bảy chữ số (gọi là phần đầu của mã nhân viên) và một chữ cái in hoa nối phía sau. Chữ cái này được tính toán từ bảy chữ số phần đầu của mã theo các bước như sau:

1. Chữ số thứ i được gắn một số nguyên dương ai được gọi là trọng lượng của nó (thứ tự các chữ số trong phần đầu của mã tính từ trái qua phải). Người ta lấy lần lượt các chữ số trong mã nhân với trọng lượng tương ứng của nó.

2.Cộng tất cả các kết quả.

3. Tìm số dư của phép chia tổng trên cho số nguyên dương K (1≤ K ≤ 11)

4. Mỗi số dư tương ứng với một chữ cái in hoa. Chữ cái in hoa tương ứng tìm được là chữ cái nối phía sau của mã nhân viên.

**Yêu cầu**: Cho trước phần đầu của mã nhân viên, hãy tìm chữ cái in hoa nối phía sau.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu vào:**   * Dòng đầu tiên ghi số nguyên dương K. * Dòng thứ 2 ghi một xâu gồm 7 chữ số là phần đầu của mã nhân viên * Dòng thứ 3 ghi 7 số nguyên dương, số thứ I là trọng lượng của chữ số thứ I trong phần đầu của mã. Các số cách nhau một ký tự trắng. * Dòng cuối cùng ghi một xâu có độ dài K gồm các chữ cái tiếng Anh in hoa, chữ cái thứ i tương ứng với số i - 1 là số dư của phép chia của tổng tìm được cho K.   **Kết quả:** chữ cái tìm được. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 11  6830907  2 7 6 5 4 3 2  JABCDEFGHIZ | D | |

#### Xe điện TRAM 116A



Nước Anh có đúng một đường xe điện. Có *n* trạm dừng, đánh số từ *1* đến *n* theo thứ tự di chuyển của xe. Tại trạm thứ *i* có *ai* hành khách xuống xe, sau đó có *bi* hành khách lên xe. Xe không có ai trước khi nó đến trạm đầu tiên và khi đến trạm cuối cùng tất cả các hành khách đều xuống xe.

Nhiệm vụ của bạn là tính sức chứa tối thiểu của xe để bất cứ thời điểm nào số lượng người bên trong xe không bao giờ vượt quá sức chứa này. Lưu ý rằng ở mỗi trạm dừng tất cả các hành khách trên xe xuống trước rồi các hành khách ở dưới mới lên xe.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input**   * Dòng đầu tiên chứa một số duy nhất *n*  - số lượng các trạm dừng của xe điện. *(2 ≤ n ≤ 1000)* * *n* dòng tiếp theo, mỗi có chứa hai số nguyên *ai* và *bi* *(0 ≤ ai, bi ≤ 1000)* - số lượng hành khách xuống xe và lên xe tại trạm thứ *i*.   *Theo đề bài ta có a1 = bn = 0. (Trạm đầu không ai xuống, trạm cuối không ai lên nữa)*  **Output**: In một số nguyên duy nhất biểu thị sức chứa tối thiểu của xe điện. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 4  0 3  2 5  4 2  4 0 | 6 | |

#### Tích lớn nhất TICHMAX NK2007

Cho một dãy gồm N số nguyên. Hãy tìm 3 số trong dãy với tích T của chúng là lớn nhất.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu**:   * Dòng đầu ghi số N (3N10000). * Dòng thứ hai chứa N số nguyên có giá trị tuyệt đối không vượt quá 30000.   **Kết quả**: Ghi ra một số duy nhất T. | **Ví dụ:**   |  |  | | --- | --- | | **Input** | **Output** | | 9  3 5 1 7 9 0 9 -3 10 | 810 | |

##### Giải

Nếu tích lớn nhất của 3 phần tử bao gồm:

* Ba số dương: ba số này phải lần lượt là số lớn nhất, nhì, ba trong dãy
* Một số âm và hai số dương: dãy phải gồm đúng hai số dương, vì nếu không ta có thể lấy tích ba số dương để đạt giá trị lớn hơn. Ta cũng suy ra ba số này phải là ba số lớn nhất, nhì, ba trong dãy.
* Hai số âm và một số dương: số dương phải là số lớn nhất và hai số âm phải là hai số nhỏ nhất, nhì trong dãy.
* Ba số âm: dãy phải gồm toàn số âm, vì nếu có một số dương ta cũng có thể lấy tích của số dương đó và hai số âm để thu được tích dương. Ta cũng suy ra được ba số cần tìm phải là ba số lớn nhất, nhì, ba của dãy.

Vậy suy ra, T=max(max1\*max2\*max3,min1\*min2\*max1), với max1, max2, max3, min1, min2 lần lượt là số lớn nhất, nhì, ba và số nhỏ nhất, nhì của dãy.

Ta có thể đọc qua lần lượt các số của dãy và cập nhật max1, max2, max3, min1, min2 mà không cần lưu lại dãy số. Đoạn lệnh dưới đây cập nhật min1, min2 khi đọc vào một số mới x:

if (x<=min1) then

begin

min2:=min1; {min1 giờ trở thành số bé thứ hai}

min1:=x; {số bé nhất là x}

end else if (x<=min2) then

min2:=x; {số bé thứ hai là x}

Việc cập nhật max1, max2, max3 được thực hiện hoàn toàn tương tự.

Chú ý vì kết quả có thể vượt quá kiểu số nguyên 32 bit nên ta cần khai báo với kiểu số nguyên 64 bit (trình biên dịch Free Pascal hỗ trợ kiểu số nguyên 64 bit với tên gọi *int64*):

var

...

kq:int64;

...

kq:=max(max1\*max2\*max3,min1\*min2\*max1);

#### [Lỗ hổng chữ số](http://nkth.bienhoanet.com/index.php?option=com_content&view=article&id=247:l-hng-ch-s&catid=28:cha-xac-nh&Itemid=33) LOHONG



Lớp của Bờm tuy không phải là lớp chuyên nhưng phong trào học Toán – Tin rất sôi động. Thầy giáo của Bờm rất tâm huyết, luôn tìm các bài toán hay, mới, lạ, đặc biệt là bài toán quy luật để dạy trò. Một hôm, thầy giáo đến lớp thật sớm trước tiết toán, viết lên bảng một dãy có quy luật sau:

42 🡪 1,  1337 🡪 0, 669 🡪 3, 1882 🡪 4, 688 🡪 5, 12345 🡪 1, 123 🡪 0, 456 🡪 2, 789 🡪 3. Và thầy đố cả lớp rằng: 45678 🡪 ?

Bằng một cái đầu rất nhạy bén Toán Tin, nhất là những bài toán Tin logic, Bờm đã kiếm được lời giải trên … Google ngay khi về đến nhà. Chật vật một hồi, cậu đã tìm ra quy luật của bài toán: chuyển số “lỗ hổng” trong các chữ số của số đã cho và biểu diễn chúng (không có chữ số 0 ở đầu). Chữ số 1, 2, 3, 5 và 7 không có lỗ hổng nào; các chữ số 0, 4, 6, 9 có một “lỗ hổng” và đặc biệt chữ số 8 có đến 2 lỗ hổng. Hôm sau Bờm rất tự tin mở rộng bài toán bằng cách thêm vào một vài số nữa đến đố bạn các lớp bên cạnh. Bạn đọc hãy giúp các bạn ấy giải bài này để Bờm không được dịp “kiêu” nhé.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dữ liệu vào :** -gồm một số nguyên duy nhất n (1 <= n <=1000000)  **Dữ liệu ra:** In ra màn hình một số nguyên duy nhất là số lỗ hổng của số đã cho. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 42 | 1 | | 669 | 3 | | 456 | 2 | | 45678 | 4 | |

##### Giải

Ta dùng mảng lohong để lưu lại số lượng lỗ hổng trên từng chữ số, số lượng này là cố định.

Const

lohong:array[0..9] of longint = (1,0,0,0,1,0,1,0,2,1);

Sau đó, tách từng chữ số của n, với mỗi chữ số ta tra trong mảng lohong để tìm số lượng lỗ hổng trong chữ số đó.

While n>0 do

Begin

tong:=tong+lohong[n mod 10];

n:=n div 10;

End;

#### VBGRASS Bãi cỏ ngon nhất



Bessie dự định cả ngày sẽ nhai cỏ xuân và ngắm nhìn cảnh xuân trên cánh đồng của nông dân John, cánh đồng này được chia thành các ô vuông nhỏ với R (1 <= R <= 100) hàng và C (1 ≤ C ≤ 100) cột. Bessie ước gì có thể đếm được số khóm cỏ trên cánh đồng.

Mỗi khóm cỏ trên bản đồ được đánh dấu bằng một ký tự ‘#‘ hoặc là 2 ký tự ‘#’ nằm kề nhau (trên đường chéo thì không phải). Cho bản đồ của cánh đồng, hãy nói cho Bessie biết có bao nhiêu khóm cỏ trên cánh đồng.

**Dữ liệu**

* Dòng 1: 2 số nguyên cách nhau bởi dấu cách: R và C
* Dòng 2..R+1: Dòng i+1 mô tả hàng i của cánh đồng với C ký tự, các ký tự là ‘#’ hoặc ‘.’ .

**Kết quả:** Một số nguyên cho biết số lượng khóm cỏ trên cánh đồng.

**Ví dụ**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Giải thích** |
| 5 6  .#....  ..#...  ..#..#  ...##.  .#.... | 5 | Cánh đồng này có 5 khóm cỏ: một khóm ở hàng đầu tiên, một khóm tạo bởi hàng thứ 2 và thứ 3 ở cột thứ 2, một khóm là 1 ký tự nằm riêng rẽ ở hàng 3, một khóm tạo bởi cột thứ 4 và thứ 5 ở hàng 4, và một khóm cuối cùng ở hàng 5. |

##### Giải

Chú ý cách đọc kí tự

Cho bảng Ar\*c, đếm xem có bao nhiêu nhóm 1 hoặc 2 kí tự # kề nhau.

Duyệt qua từng kí tự trong bảng

* if a[I,j] = ‘#’ then inc(dem);
* if a[I, j-1]= ‘#’ then dec(dem);
* if a[i-1, j]=’#’ then dec(dem);

#### Ngôn ngữ LINGVO

Steve học ngoại ngữ. Bài tập hôm nay là học thuộc tên các chữ số (ở hệ cơ số 10). Để rèn luyện kỹ năng phản xạ nhanh. Steve viết một dãy *n* số nguyên dương *a1, a2, …, an*, mỗi số không vượt quá *109*và không có các số *0* không có nghĩa ở đầu. Với mỗi số Steve sẽ đọc chữ số có tên lớn nhất (theo thứ tự từ điển) trong số các tên xuất hiện trong số này.

**Ví dụ**, Steve học tiếng Pháp. Tên các chữ số trong tiếng Pháp là như sau:



Với số *908*, chữ số mà Steve đọc là *zero*.

Yêu cầu: cho dãy *10* tên các chữ số từ *0* đến *9*, số nguyên n và n số nguyên *a1, a2, …, an.* Với mỗi số nguyên hãy nêu tên chữ số được đọc.

**Input**

* Dòng đầu tiên chứa 10 xâu, xâu thứ i là tên chữ số i, i = 0 .. 9, mỗi xâu không quá 50 ký tự, các xâu cách nhau một dấu cách.
* Dòng thứ 2 chứa số nguyên *n*
* Dòng thứ i trong n dòng tiếp theo chứa số nguyên *ai*.

**Output**: n tên các chữ số được đọc, mỗi tên đưa ra trên một dòng, dòng thứ i xác định tên đọc trong số ai.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| zero un deux trois quatre cinq six sept huit neuf  3  123  456  908 | un  Six  zero |

##### Giải

Tách từng chữ số

Tìm chữ số lớn nhất theo cách đọc.

#### Chữ số thứ N CHUSON

Khi viết các số tự nhiên tăng dần từ 1, 2, 3,… liên tiếp nhau, ta nhận được một dãy các chữ số thập phân vô hạn, ví dụ: 1234567891011121314151617181920...

**Yêu cầu:** Hãy tìm chữ số thứ N của dãy số vô hạn trên.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu vào:** gồm một số dòng, mỗi dòng ghi một số nguyên dương N (N<109).  **Kết quả ra:** với mỗi số N, ghi trên dòng tương ứng chữ số thứ N của dãy. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 5  10  54 | 5  1  3 | |

##### Giải

##### Cách 1

Cho biến i tăng dần, ta đến số chữ số của i cho đến khi đủ n chữ số.

d:=0;

while d<n do

begin

x:=i;

while x>0 do

begin

x:=x div 10;

inc(dem);

end;

end;

vì mỗi lần ta cộng thêm một số lượng chữ số chữ số của i, nên khi vòng lặp ngoài dừng lại có thể vượt quá n. Ví dụ với n = 11, khi vòng lặp trên dừng lại d sẽ bằng 12. Ta tính số lượng chữ số dư x và quay ngược lại để tìm ra chữ số thứ n.

x:=d-n;

for i:=1 to x do i:=i div 10;

writeln(fo,i);

##### Cách 2

Gọi num là số mà chứa chữ số thứ n. scs là số chữ số của num. Đầu tiên ta tìm xem num là số có mấy chữ số. Nhận xét:

Từ 1 -> 9 có 9 chữ số

Từ 10 -> 99 có 9 + 2\*9\*10 chữ số

Từ 100 -> 999 có 3\*9\*100 chữ số

Ta tăng dần các lớp chữ số để tìm số chữ số của x.

Sum := 9; Num := 1; Scs := 1;

while Sum < n do

begin

Num := Num \* 10; Inc(scs);

Sum:=Sum + Scs\*9\*Num;

end;

Bước 2, ta tìm số chữ số trong dãy ứng với lớp chữ số scs, cập nhật lại biến n

Dec(Sum, Num \* 9 \* Digits); //so chu so den het lop scs-1

Dec(n, Sum);//so chu so lop scs

Bước 3, tìm số num (có thể chưa hết số)

Num := Num + (n - 1) div Digits;

n := (n - 1) mod Digits + 1;

for i := 1 to Digits - n do Num := Num div 10;

Bước 4, lấy chữ số cuối cùng của Num làm kết quả

Result := Num mod 10;

#### May quốc kì QUOCKY



Gọi x, y là chiều dài và chiều rộng lá quốc kì, ta có:   
Người ta may N lá quốc kì, chiều dài lá thứ i bằng chiều rộng lá thứ i-1.  
Cho biết chiều rộng y của lá thứ K trong số N lá. Tính chiều dài và chiều rộng của lá lớn nhất và nhỏ nhất.

**Input**: gồm 3 số n, k, y

**Output**: kết quả

* Dòng đầu là chiều rộng và chiều dài của lá nhỏ nhất, làm tròn 5 chữ số.
* Dòng hai là chiều rộng và chiều dài của lá lớn nhát, làm tròn 5 chữ số.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 5 100 | 9.01699 14.58980  685.41020 1109.01699 |

##### Giải

Lá lớn nhất là thứ 1, lá nhỏ nhất là lá n.

Để đơn giản biểu thức ta Đặt u = .

Const u = (sqrt(5) – 1)/2;// y=ux // x = y/u

Ta có mối liên hệ giữa x và y là y = u\*x và x = y/u tức là có chiều dài sẽ tính được chiều rộng và ngược lại.

Ta lặp từ k+1 lên n để tìm lá thứ n. Ta chỉ sử dụng 2 biến x, y để lưu chiều dài và rộng của các lá thứ k+1, k+2, ..., n

Mỗi lần lặp ta cập nhật chiều dài x, chiều rộng y của lá cờ.

//y dang la chieu rong cua la thu k

//tim la nho nhat

for i:=k+1 to n do

begin

x:=y;// chiều dài lá thứ i bằng chiều rộng lá trước đó

y:=u\*x;

end;

Tương tự, Ta lặp từ k về 1 để tìm lá lớn nhất

#### Biến đổi dãy số BIENDOI

Cho dãy các số nguyên khác nhau đôi một a gồm n số a1, a2, ..., an và dãy số nguyên b gồm n số b1, b2, ..., bn. Trên dãy số a ta có thể áp dụng phép biến đổi T(i) thực hiện phép hoán vị giá trị hai phần tử ai và ai+1 (0<i<n) .Vấn đề đặt ra là có tồn tại hay không một dãy các phép biến đổi Ti1, Ti2, …, Tik sao cho khi áp dụng, dãy a ban đầu sẽ biến thành dãy b.

**Ví dụ** nếu dãy a là 1, 3, 2, 4 và dãy b là 2, 1, 3, 4 thì ta có thể sử dụng dãy 2 phép biến đổi T(2) và T(1) để biến đổi a thành b.

**Yêu cầu:** Cho hai dãy số a, b. Hãy chỉ ra một dãy các phép biến đổi a thành b hoặc cho biết không tồn tại dãy biến đổi như vậy.

**Dữ liệu:** gồm 3 dòng:

* Dòng đầu tiên chứa số nguyên n là số lượng phần tử của mỗi dãy số (n<=100).
* Dòng thứ hai chứa n số nguyên đôi một khác nhau ứng với các phần tử của dãy số a.
* Dòng cuối cùng chứa n số nguyên ứng với các phần tử của dãy số b. (-10000 ≤ ai, bi ≤ 10000)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kết quả:** Nếu không thể biến đổi a thành b, file chứa số duy nhất số -1. Trong trường hợp ngựoc lại, file sẽ gồm 2 dòng:   * + Dòng đầu tiên chứa số nguyên k là số lượng phép biến đổi cần áp dụng.   + Dòng thứ hai chứa k số nguyên duơng i1, i2, …, ik ứng với các phép biến đổi Ti1, Ti2, …, Tik tìm được. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 4  1 3 2 4  2 1 3 4 | 2  2 1 | | 1  5  2 | -1 | |

##### Giải

Thuật toán của ta đơn giản như sau: tìm b[1] trong dãy a rồi di chuyển lên đầu dãy a, sau đó lại tìm b[2] rồi di chuyển lên vị trí thứ hai của dãy a, ... Nếu tại một bước ta không tìm thấy số tương ứng trong dãy a thì in ra -1.

Đoạn chương trình sau thể hiện thuật toán:

for i:=1 to n do {duyệt qua tất cả các phần tử của dãy b}

begin

vt[i]:=0; {vt[i] là vị trí xuất hiện của b[i] trong dãy a}

for j:=i to n do {duyệt qua các phần tử của dãy a để tìm số b[i]}

if (a[j]=b[i]) then {tìm đặt b[i]}

begin

vt[i]:=j; {cập nhật lại vt[i]}

break; {thoát khỏi vòng lặp}

end;

if (vt[i]<>0) then {nếu tìm được số b[i]}

begin

kq:=kq+vt[i]-i; {cần thực hiện thêm vt[i]-i phép biển đổi

để đưa số từ vị trí vt[i] về vị trí i}

for j:=vt[i]-1 downto i do trao(j) {thực hiện biến đổi!}

end

else

begin {nếu không tìm được số b[i]}

writeln(-1); {in ra -1}

exit; {thoát khỏi thủ tục}

end;

end;

#### Robot ROBOT

*2012\_HSG12\_Đồng Nai\_V1*

Trong cuộc thi thiết kế robot khối THPT, ban tổ chức yêu cầu các đội thiết kế robot và lập trình cho phép robot di chuyển qua các ô kề cạnh trong một bảng kẻ ô vuông n dòng, m cột. Các dòng đánh số từ 1 đến n theo thứ tự từ trên xuống và các cột đánh số từ 1 đến m theo thứ tự từ trái sang phải. Trong bảng có một số ô được tô đen.

Robot bắt đầu di chuyển từ ô dòng 1, cột 1, không được di chuyển vào các ô đen và không được ra khỏi bảng, các đội ghi lại lộ trình robot đã đi theo các kí hiệu như sau:

* U: di chuyển lên trên một ô
* D: di chuyển xuống dưới một ô
* L: di chuyển sang trái một ô
* R: di chuyển sang phải một ô

**Ví dụ**, lộ trình di chuyển là RRDDLRRU – tử ô (1, 1) di chuyển sang phải 2 ô, xuống dưới 2 ô, sang trái 1 ô, sang phải 2 ô và cuối cùng lên trên một ô.

**Yêu cầu**: Hãy xác định vị trí đến của robot.

**Input**

* Dòng đầu tiên là 3 số nguyên n, m, k (2 ≤ n, m ≤ 1000, k<104)
* K dòng tiếp theo, mỗi dòng gồm hai số i và j, vị trí dòng và cột của ô được tô đen.
* Dòng cuối là một xâu kí tự ghi lộ trình của robot đã thực hiện. Độ dài xâu không quá 104.

**Output:** vị trí dòng, cột của ô cuối cùng trên lộ trình. Nếu lộ trình của robot vi phạm luật chơi thì xuất ra hai số -1 -1

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Giải thích** |
| 4 4 1  4 2  RRDDLRRU | 2 4 |  |

##### Giải BANG

Đánh dấu ô tô đen có a[i, j] = -1

Giả sử ô hiện tại là a[i, j]. đọc từng kí tự trong xâu đường đi và tăng giảm (i , j) để đi qua ô kế tiếp.

Có thể tạo biên là các ô có gt -1 để tránh đi ra ngoài.

#### Các quân bài CARD



Ban đầu mụ phù thủy sẽ xếp N quân bài khác nhau thành một hàng ngang trên mặt bàn. Tiếp theo mụ sẽ làm M lần việc đổi chỗ 2 quân bài nào đó trong dãy bài. Cụ thể: lần thứ i mụ sẽ đổi chỗ quân bài thứ Xi và Yi (tính từ trái sang phải).

**Yêu cầu**: bạn phải chỉ ra quân bài thứ nhất trong dãy bài ban đầu bây giờ nằm ở đâu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input**   * Dòng đầu gồm 2 số nguyên N và M (1 ≤ N ≤ 109, 0 ≤ M ≤ 105) * M dòng sau, dòng thứ i chứa 2 số nguyên phân biệt Xi và Yi (1 ≤ Xi, Yi ≤ N)   **Output:** Gồm một số nguyên duy nhất là vị trí hiện tại của quân bài thứ nhất. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | 10 2  1 3  3 4 | 4 | |

##### Giải

Chúng ta sẽ dùng biến P để lưu vị trí hiện tại của quân bài thứ nhất trong dãy bài lúc đầu. Duyệt qua lần lượt các phép đổi chỗ quân bài Xi và Yi:

* Nếu Xi <> P và Yi <> P thì P không đổi.
* Nếu Xi = P thì P = Yi
* Nếu Yi = P thì P = Xi

Câu lệnh cho mỗi lần đổi bài như sau:

if p = x then p:=y

else

if p = y then p:=x;

chú ý ở đây là không cần xét trường hợp 1 vì P không đổi thì không làm gì cả. Ta cũng không thể dùng 2 câu lệnh if như sau

if p = x then p:=y

if p = y then p:=x;

vì trong 2 trường hợp này, chỉ có nhiều nhất 1 trường hợp được xảy ra.

#### Nhà sưu tầm đồ cổ DOCO



Đầu tiên, bé Nam chỉ có 1 món đồ với “độ tuổi” là 1 ngày. Trong N ngày tiếp theo, ngày thứ i, cậu ghi lại tuổi của món đồ cổ nhất mà mình có (món đồ có độ tuổi “lớn nhất”), sau đó cậu sẽ bổ sung thêm một đồ vật có độ tuổi xi (ngày) vào bộ sưu tập của mình. Công việc tưởng chừng đơn giản nhưng khi số lượng đồ cổ tăng lên và đặc biệt là việc sau mỗi ngày độ tuổi của món đồ cổ sẽ tăng lên 1, bé Nam thấy không thể tính toán nhanh được. Bạn hãy giúp bé Nam nhé!

Chú ý: nếu ngày i Nam nhận được món đồ cổ có tuổi là x thì đến hết ngày i, món đồ cổ này sẽ tăng lên thành x + 1 tuổi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input**   * Dòng đầu gồm 1 số nguyên duy nhất là N (N ≤ 103) * N dòng sau, dòng thứ i ghi xi (xi ≤ 103)   **Output:** Gồm n dòng, dòng i ghi lại câu trả lời tương ứng với ngày thứ i | **Ví dụ:**   |  |  | | --- | --- | | **Input** | **Output** | | 4  3  1  10  25 | 2  4  5  11 | |

##### Giải

Ta không cần quan tâm đến từng món đồ cổ mà Nam có, chỉ cần quan tâm đến món đồ có độ tuổi lớn nhất. Gọi A là tuổi món đồ cổ nhất của Nam. Sau mỗi ngày, giá trị của A sẽ tăng lên 1. Khi thêm một món đồ cổ độ tuổi B, ta sẽ cập nhật lại giá trị A. Cụ thể A = max(A, B)

#### Màu áo MAUAO 268A

Lê làm việc ở kênh thể thao của đài truyền hình VTV. Anh ấy thường xuyên xem bóng đá các nước. Khi xem bóng đá, anh ấy thường chú ý màu áo của mỗi đội. Anh ấy nhận ra rằng, mỗi đội có hai màu áo, màu sân nhà và màu sân khách. Khi đội chơi ở sân nhà, các cầu thủ sẽ mặc áo màu sân nhà, ngược lại khi chơi trên sân khách sẽ mặc áo màu sân khách. Có 1 ngoại lệ: khi màu áo sân nhà của đội nhà trùng với màu áo sân khách của đội khách, đội chủ nhà phải mặc áo màu sân khách. Đồng phục sân nhà và đồng phục sân khách phải khác màu.

Có n đội tham gia giải vô địch quốc gia. Giải gồm n\*(n-1) trận đấu: mỗi đội sẽ gặp các đội khác trên sân của họ. Lúc này, Lê tự hỏi: có bao nhiêu lần trong suốt giải đấu: đội chủ nhà sẽ phải mặc đồng phục sân khách? Chú ý là thứ tự trận đấu sẽ không ảnh hưởng đến số lần này.

Bạn biết được màu đồng phục sân nhà và sân khách của mỗi đội. Các màu được biểu diễn bởi các số nguyên. Hai màu khác nhau khi giá trị số khác nhau. Bạn hãy giúp Lê tìm ra câu trả lời nhé!

**Input**

* Dòng đầu: số n (2 ≤ n ≤ 30)
* N dòng sau: mỗi dòng chứa 2 số nguyên khác nhau hi, ai – màu áo sân nhà và sân khách của đội thứ i tương ứng. ( 1 ≤ hi, ai ≤ 100)

**Output**: số lượng trận đấu mà đội chủ nhà phải mặc áo sân khách.

**Sample test(s)**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** |  |
| 3 1 2  2 4  3 4 | 1 | Giải vô địch có 6 trận, có một trận thỏa yêu cầu đề bài là trận giữa đội 2 và đội 1 trên sân đội 2. |
| 4 100 42 42 100 5 42 100 5 | 5 |  |
| 2 1 2 1 2 | 0 |  |

##### Giải

Chú ý là thứ tự trận đấu sẽ không ảnh hưởng đến số lần này.

Chỉ với 30 đội, cách đơn giản nhất là duyệt qua tất cả các trận đấu:

for i = 1 to N

for j = 1 to N

if i <> j and h[i] = a[j] then ++ans

##### Cách 2

Với mỗi màu áo i, đếm số đội cntH[i] có màu áo sân nhà là màu i, cntA[i] là số đội có màu áo sân khách là i. Kết quả là tổng các tích cntH[i]\*cntA[i] với i=1..m.

#### Xâu đối xứng XAUDX

*Đề thi học sinh giỏi 12 tỉnh Đồng Nai năm 2011 vòng 1*

Cho xâu X độ dài không quá 1000, chỉ chứa các kí tự trong tập {‘?’, ‘A’, ..., ‘Z’}. Trong đó, mỗi kí tự ‘?’ có thể được thay bằng một trong các kí tự ‘A’, ..., ‘Z’. Ví dụ xâu ‘A?DA’, ta có thể tạo được các xâu ‘AADA’, ‘ABDA’, ‘ACDA’, ‘ADDA’, ... ‘AZDA’. Trong đó xâu ‘ADDA’ là xâu đối xứng.

**Yêu cầu**: cho xâu X. Hãy tìm xâu đối xứng có thứ tự từ điển nhỏ nhất có thể tạo được từ X.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu vào**: xâu X.  **Kết quả**: xâu tìm được hoặc ghi ra -1 nếu không tìm được. | **Ví dụ**   |  |  | | --- | --- | | **Input** | **Output** | | DF????D  AKD? | DFAAAFD  -1 | |

##### Giải

Duyệt từ kí tự đầu đến nửa đầu của xâu, xét từ cặp kí tự đối xứng s[i] = x và s[n+1-i] = y. Có 3 khả năng xảy ra:

* nếu x và y là 2 chữ cái khác nhau thì xâu không thể đối xứng
* nếu x= ? và y <> ? then y = x và ngược lại
* nếu x=? và y=? then x=y=’A’ để được xâu đối xứng nhỏ nhất