**BÀI TẬP ĐỘI DỰ TUYỂN**

**Bài 1. Đếm vùng.**

Cho ma trận kích thước m x n (2<=m,n<=20) các giá trị của ma trận là số nguyên dương (<= 105) hãy đếm các vùng là số nguyên tố của ma trận *(các số nguyên tố được coi là cùng một vùng nếu chúng nằm cạnh nhau theo hướng ngang hoặc dọc)*?

Dữ liệu vào file DEMVUNG.INP dòng đầu là 2 số m và n cách nhau một dấu cách m dòng tiếp theo là các giá trị của ma trận.

Kết quả ra file DEMVUNG.OUT số vùng nguyên tố của ma trận trên.

Ví dụ:

|  |  |
| --- | --- |
| DEMVUNG.INP | DEMVUNG.OUT |
| 5 5  2 3 4 5 3  5 6 8 7 2  3 7 6 5 3  3 4 5 7 4  2 7 6 9 3 | 3  0 0 4 0 0  0 6 8 0 0  0 0 6 0 0  0 4 0 0 4  0 0 6 9 0 |

**Bài 2.** Mua vé tàu hỏa

Tuyến đường sắt từ thành phố A đến thành phố B đi qua một số nhà ga. Có thể biểu diễn như là các điểm trên đoạn thẳng AB. Nhà ga A có số hiệu 1, nhà ga tiếp theo là 2,... nhà ga cuối cùng tại B có số hiệu N. Giá vé đi thẳng giữa hai nhà ga nào đó phụ thuộc vào khoảng cách giữa chúng. Cách tính giá vé như sau: Gọi khoảng cách giữa hai ga là x

* Nếu 0≤x≤L1 thì giá vé là C1
* Nếu L1<x≤L2 thì giá vé là C2
* Nếu L2<x≤L3 thì giá vé là C3

Vé đi thẳng từ nhà ga này đến nhà ga khác chỉ có thể đặt mua nếu khoảng cách giữa chúng không vượt quá L3. Vì thế nhiều khi để đi từ ga này đến ga khác ta phải mua một số vé. Ví dụ, trên tuyến đường sắt đã cho, giữa ga 2 và ga 6 là khoảng cách 2\*L2>L3. Rõ ràng ta không thể mua vé đi thẳng, như có nhiều cách giải quyết. Chẳng hạn mua vé đi từ ga 2 đến ga 3 mất chi phí C2 và mua vé từ ga 3 đi ga 6 mất chi phí C3 và chi phí tổng cộng để mua vé từ ga 2 đến ga 6 mất C2+C3. Lưu ý là mặc dù khoảng cách từ ga 2 đến ga 6 là 2\*L2 nhưng không thể đặt mua 2 vé với giá C2 để đi từ ga 2 đến ga 6 với chi phí tổng cộng 2\*C2 vì mỗi vé chi có giá trị đi lại giữa hai nhà ga tương ứng.

Yêu cầu tìm cách đặt mua vé để đi lại giũa hai nhà ga cho trước với chi phí mua vé nhỏ nhất.

**Dữ liệu** vào từ file văn bản RTICKET.INP

* Dòng đầu tiên ghi các số L1, L2, L3, C1, C2, C3
* Dòng thứ 2 ghi số lượng nhà ga N
* Dòng thứ 3 ghi 2 số nguyên là số hiệu nhà ga cần mua vé: s và t
* N-1 dòng tiếp theo, mỗi dòng thứ i trong N-1 dòng này ghi số nguyên là khoảng cách từ nhà ga 1 (tại A) đến nhà ga i+1 (i=1,2,...,N-1)

**Kết quả** ghi ra file RTICKET.OUT chi phí nhỏ nhất tìm được (biết rằng luôn có cách mua vé để đi lại giữa hai ga bất kỳ)

Hạn ché kỹ thuật: 1≤L1<L2<L3≤109, 1≤C1<C2<C3≤109, 1≤N≤104

*Ví dụ:*

|  |  |
| --- | --- |
| RTICKET.INP | RTICKET.OUT |
| 3 6 8 20 30 40  7  2 6  3  7  8  13  15  23 | 70 |

**Bài 3. NỐI ĐIỂM**

Trên hai đường thẳng song song L1 và L2, người ta đánh dấu trên mỗi đường N điểm. Các điểm trên đường thẳng L1 được đánh số 1, 2, …, N từ trái qua phải, còn các điểm trên đường L2 được đánh số bởi D[1], D[2], …, D[n] là một hoán vị của N, cũng được đánh dấu từ trái qua phải (hình vẽ dưới đây cho một ví dụ khi N = 9)

L1 1---2---3---4---5---6---7---8---9

L2 2---5---3---8---7---4---6---9---1

Với số nguyên không âm M cho trước, ta được phép nối hai điểm thứ i trên L1 với điểm thứ j trên L2 nếu: ***ABS(i-D[j]) ≤ M***

***Yêu cầu:*** Tìm cách nối được nhiều cặp điểm nhất với điều kiện các đoạn nối không được cắt nhau.

***Dữ liệu vào trong file: “noidiem.inp”***

* Dòng đầu tiên chứa số nguyên N và M (0≤M<N≤1000).
* Dòng thứ 2 chứa các số D[1], D[2], …, D[n].

***Kết quả ra file “noidiem.out”*** Ghi số K là số lượng cặp điểm nối tìm được.

|  |  |
| --- | --- |
| noidiem.inp | noidiem.out |
| 3 1  2 1 3 | 3 |

**Bài 4. Trả tiền**

Một ngân hàng có N loại tiền mênh giá A[1], A[2],.., A[N] với số lượng tiền mỗi loại không giới hạn. Cần chi trả cho khách hàng một số tiền là M đồng. Hãy cho biết cần bao nhiêu tiền mỗi loại để chi trả sao cho số lượng tờ là ít nhất.

Cho biết: N ≤ 100; A[i] < 256; M ≤ 10000.

Dữ liệu vào file **Tratien.inp:** dòng đầu là 2 số N và M, dòng 2 ghi N số nguyên đương A[1], A[2],.., A[N].

Dữ liệu ra file **Tratien.out**: dòng đầu ghi số lượng phải trả, dòng thứ 2 ghi N số nguyên không âm ứng với số tờ cần trả cho mỗi loại tiền.

Ví dụ:

|  |  |
| --- | --- |
| TRATIEN.INP | TRATIEN.OUT |
| 5 97  1 5 10 50 100 | 8  2 1 4 1 0 |

**Bài 5. BỘI SỐ**

Bội của số X là số chia hết cho X. Viết chương trình đếm số bội của X nằm trong đoạn [A, B] sao cho thỏa mãn ở trong hệ thập phân nó chi chứa các chữ số cho phép.

**Dữ liệu vào từ file V.INP**

Dòng đầu tiên chứa 3 số nguyên X, A, B (1<X< 1011, 1<A<B<1011).

Dòng thứ 2 chứa các chữ số cho phép. Các chữ số được cho không ngăn cách bởi dấu cách, theo thứ tự tăng dần và không lặp lại.

**Kết quả ghi ra file V.OUT**

Chỉ một số duy nhất là số bội tìm được.

Ví dụ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **V.INP** | **V.OUT** | **V.INP** | **V.OUT** | **V.INP** | **V.OUT** |
| 2 1 20 0123456789 | 10 | 6 100 9294 23689 | 111 | 5 4395 9999999999 12346789 | 0 |

**Bài 6.DẤU NGOẶC**

Một chuỗi dấu ngoặc được gọi là chuỗi đúng nếu nó thỏa mãn các điều kiện sau:

* Chuỗi rỗng là chuỗi đúng.
* Nếu A là chuỗi đúng thì(A), [A] và {A} cũng là chuỗi đúng.
* Nếu A và B là chuỗi đúng thì AB là chuỗi đúng.

Ví dụ các chuỗi [({})]; [](){} ; [{}]()[{}] là các chuỗi đúng. Còn các chuỗi [({{([; []({)} và [{}])([{}] không phải là chuỗi đúng.

Tuấn có một chuỗi dấu ngoặc là chuỗi đúng, nhưng có một số dấu ngoặc đã bị nhòe không đọc được, hãy viết chương trình tính toán xem có bao nhiêu cách thay thế các dấu ngoặc không đọc được trong chuỗi thành các dấu ngoặc để được chuỗi đúng.

Dữ liệu vào từ file**DAUNGOAC.INP**:

* Dòng đầu tiên chứa 1 số nguyên dương chẵn N (2< N < 200).
* Dòng thứ 2 chứa 1 chuỗi, các dấu ngoặc không đọc được trong chuỗi được thay bằng dấu ?.

Kết quả ghi ra file **DAUNGOAC.OUT**:

Chỉ 1 số duy nhất là số chuỗi đúng được tạo thànhtừ chuỗi đã cho. Số này có thể rất lớn nên chỉ đưa ra 5 chữ số cuối cùng của số tìm được.

*Ví dụ:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DAUNGOAC.INP** | **DAUNGOAC.OUT** | **DAUNGOAC.INP** | **DAUNGOAC.OUT** | **DAUNGOAC.INP** | **DAUNGOAC.OUT** |
| 6  ()()() | 1 | 10  (?([?)]?}? | 3 | 16  ???[???????]???? | 92202 |