1. **Cây gỗ quý Tên file chương trình ‘TIMBER.???’**

Một khu đất rộng lớn có thể xem như một mặt phẳng tọa độ Oxy, người ta trồng trên khu đất đó n cây gỗ quý, cây gỗ quý thứ i được trồng ở tọa độ nguyên (xi,yi). Để bảo vệ các cây gỗ quý vào ban đêm, người ta lắp đặt một cái đèn chiếu sáng trên không chiếu xuống khu đất này. Biết rằng đèn chiếu sáng này có thể di chuyển theo các hướng đi lên, đi xuống, qua trái, qua phải mà vẫn đảm bảo ánh sáng chiếu xuống khu đất có dạng hình vuông có cạnh là k (nguyên) và có các cạnh song song với 2 trục tọa độ.

Người ta muốn biết rằng với cái đèn chiếu sáng đó thì trong một thời điểm xác định sẽ chiếu sáng được nhiều nhất là bao nhiêu cây gỗ quý.

***Ví dụ:*** *Có 7 cây gỗ quý có tọa độ tương ứng là (3, 1); (0, 1); (4, 2); (-2, -3); (2, -2); (2, 3); (0, -2)*

* *Có nhiều nhất 3 cây gỗ quý có tọa độ tương ứng là (3, 1); (4, 2); (2, 3) được chiếu sáng trong cùng một thời điểm với ánh sáng chiếu xuống khu vườn có dạng hình vuông cạnh là 2, song song với 2 trục tọa độ.*

x

y

**•**

(2,3)

**•**

(4,2)

**•**

(3,1)

**•**

(0,1)

**•**

(2,-2)

**•**

(0,-2)

**•**

(-2,-3)

O

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Yêu cầu:** Hãy cho biết số cây gỗ quý nhiều nhất được chiếu sáng trong cùng một thời điểm. Biết rằng cây gỗ quý được chiếu sáng ngay trên cạnh hình vuông cũng được tính.

**Dữ liệu** vào từ file ‘TIMBER.INP’:

* Dòng thứ nhất chứa số nguyên dương n và k (n≤105; k≤102).
* n dòng tiếp theo, dòng thứ i chứa 2 số nguyên xi và yi (|xi|, |yi|≤103; i=1,2,…n).

**Kết quả** ghi vào file ‘TIMBER.OUT’ một số nguyên dương duy nhất là số lượng cây gỗ quý được chiếu sáng nhiều nhất trong cùng một thời điểm.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **TIMBER.INP** |  | **TIMBER.OUT** |
| 7 2  3 1  0 1  4 2  -2 -3  2 -2  2 3  0 -2 |  | 3 |

1. **Đường đi ngắn nhất Tên file chương trình ‘SORTPATH.???’**

Cho đồ thị vô hướng n đỉnh

**Yêu cầu:** Tìm đường đi ngắn nhất từ s đến t

**Dữ liệu**vào từfile ‘SORTPATH.INP’

* Dòng thứ nhất chứa các số nguyên: n, s, t (n≤1000)
* Các dòng tiếp theo mỗi dòng chứa 3 số nguyên i, j, k ; trong đó k là trọng số của cạnh i, j (k ≥ 0)

**Kết quả** ghi vào file‘SORTPATH.OUT’ chứa một số nguyên duy nhất là độ dài dường đi ngắn nhất từ s đến t.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **SORTPATH.INP** |  | **SORTPATH.OUT** |
| 4 1 4  1 2 5  1 4 12  2 3 4  2 4 1  3 5 6 |  | 6 |

1. **Đường đi Tên file chương trình ‘FINDPATH.???’**

Cho đồ thị vô hướng n đỉnh, không trọng số

**Yêu cầu:** tìm tất cả các đường đi (khác nhau) từ s đến t.

**Dữ liệu** vào từ file ‘FINDPATH.INP’:

* Dòng thứ nhất chứa 3 số nguyên: n, s, t (n≤20).
* Các dòng tiếp theo mỗi dòng chứa 2 số nguyên i, j mô tả cạnh i, j của đồ thị

**Kết quả** ghi vào file ‘FINDPATH.OUT’ chứa một số nguyên duy nhất là số đường đi khác nhau từ s đến t

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **FINDPATH.INP** |  | **FINDPATH.OUT** |
| 5 1 5  1 4  1 3  5 1  2 1  4 3  4 2  2 5 |  | 4 |

1. **Thành phần liên thông Tên file chương trình ‘CONNECT.???’**

Cho đồ thị vô hướng n đỉnh, không trọng số

**Yêu cầu:** đếm số thành phần liên thông của đồ thị.

**Dữ liệu** vào từ file ‘CONNECT.INP’:

* Dòng thứ nhất chứa 1 số nguyên: n (n≤1000).
* Các dòng tiếp theo mỗi dòng chứa 2 số nguyên i, j mô tả cạnh i, j của đồ thị

**Kết quả** ghi vào file ‘CONNECT.OUT’ chứa một số nguyên duy nhất là số thành phần liên thông của đồ thị.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **CONNECT.INP** |  | **CONNECT.OUT** |
| 7  4 2  3 1  5 1  7 3  7 4  2 5  7 5 |  | 2 |

1. **Người du lịch Tên file chương trình‘TOURIST.???’**

Có n thành phố (được đánh số từ 1 đến n). Một người du lịch xuất phát từ 1 thành phố, muốn đi thăm các thành phố khác, mỗi thành phố đúng 1 lần rồi quay lại về nơi xuất phát. Biết chi phí đi từ thành phố i đến thành phố j là C[i,j].

**Yêu cầu**: Hãy tìm 1 hành trình cho người du lịch để tổng chi phí theo hành trình này là nhỏ nhất. Giả sử là từ i đến j luôn tồn tại 1 đường đi.

**Dữ liệu** vào từ file ‘TOURIST.INP’:

* Dòng thứ 1 số nguyên n (n≤18)
* Các dòng tiếp theo, mỗi dòng chứa 3 số nguyên i, j, k (k là chi phí đường đi từ thành phố i đến j)

**Kết quả** ghi vào file ‘TOURIST.OUT’ chứa một số nguyên duy nhất là tổng chi phí nhỏ nhất của hành trình

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **TOURIST.INP** |  | **TOURIST.OUT** |
| 4  1 2 5  1 3 3  1 4 9  2 4 6  2 3 10  3 4 3 |  | 17 |

1. **Money Tên file chương trình ‘MONEY.???’**

Một người đi du lịch vòng quanh thế giới, ông ta chỉ được phép mang theo số tiền nhiều nhất là S, ông ta có N tờ giấy bạc được đánh số từ 1 tới N, tờ giấy bạc thứ i có mệnh giá ai. Muốn đủ chi phí cho chuyến đi ông ta phải mang theo số tiền là S.

**Yêu cầu:** Hãy giúp người du lịch chọn ra trong N tờ giấy bạc mà ông ta có để có được số tiền là S với số lượng các tờ giấy là ít nhất có thể.

**Dữ liệu** vào từ file ‘MONEY.INP’:

* Dòng đầu chứa 2 số nguyên dương N và S (0 ≤ N, S ≤ 1000 ).
* N dòng tiếp theo, dòng thứ i mệnh giá của tờ giấy bạc thứ i (ai ≤ 104)

**Kết quả**ghi vào file ‘MONEY.OUT’ Nếu có thể chọn được số tiền là S từ các tờ giấy bạc mà người du lịch có thì ghi ra số lượng các tờ giấy bạc ít nhất  *(Nếu không thể chọn để có được đúng số tiền là S thì ghi ra số -1)*

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **MONEY.INP** |  | **MONEY.OUT** |
| 10 390  200 10 20 20 50 50 50 50 100 100 |  | 5 |

1. **Tổng lớn nhất Tên file bài làm: ‘SUMMAXK.???’**

Cho dãy gồm N số nguyên dương a1, a2,…, an.

**Yêu cầu**: Hãy xác định K phần tử liên tiếp trong dãy có tổng lớn nhất

**Dữ liệu** vào từ file ‘SUMMAXK.INP’

* Dòng thứ nhất chứa 2 số nguyên N, K (1<N< 2.106 , 2<K<1000)
* N dòng tiếp theo, dòng thứ i chứa số nguyên ai (0<ai<1000)

Kết quả ghi vàofile ‘SUMMAXK.OUT’chứa tổng lớn nhất của Kphần tử liên tiếp trong dãy.

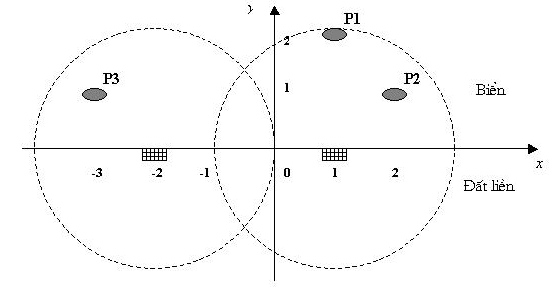
**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **SUMMAXK.INP** |  | **SUMMAXK.OUT** |
| 6 3  5  1  9  8  7  2 |  | 24 |

1. **Trạm Radar Tên file chương trình ‘RADAR.???’**

Coi như bờ biển là một đường thẳng chia mặt phẳng làm hai phần: Đất liền và biển. Mỗi hòn đảo nhỏ là một điểm nằm bên phía biển. Mỗi trạm Radar nằm trên bờ biển có khoảng phủ sóng là *d*, do đó mỗi hòn đảo trên biển có thể bị theo dõi bởi một trạm Radar nếu khoảng cách giữa chúng không quá *d*. Chúng ta sử dụng hệ toạ độ Đề các với bờ biển là *trục x*. Phần biển nằm phía trên trục x, còn phần đất liền ở phía dưới.

**Yêu cầu:** Cho vị trí các đảo và khoảng bao phủ của Radar, xác định số Radar tối thiểu cần thiết để theo dõi được tất cả các đảo. Vị trí của mỗi hòn đảo cho bởi toạ độ (x,y).



**Dữ liệu** vào từ file‘RADAR.INP’ là một bộ gồm nhiều test. Dòng đầu mỗi test gồm 2 số nguyên *n* (*n<=1000*) và *d*, trong đó *n* là số đảo trên biển, *d* là khoảng bao phủ của Radar. Mỗi dòng trong *n* dòng tiếp theo là hai số nguyên chỉ toạ độ của từng đảo. Các test cách nhau một dòng trống. Dữ liệu vào kết thúc bởi một dòng gồm hai số 0

**Kết quả** ghi vào file ‘RADAR.OUT’

Mỗi dòng là một số nguyên thể hiện số Radar tối thiểu tìm được cho mỗi test. Ghi -1 nếu không có cài đặt được

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **RADAR.INP** |  | **RADAR.OUT** |
| 3 2  1 2  -3 1  2 1  1 2  0 2  0 0 |  | 2  1 |

1. **Truyền tin Tên file chương trình ‘MESSAGES.???’**

Cần chuyển hết n gói tin trên một mạng có m kênh truyền. Biết chi phí để chuyển i gói tin trên kênh j là C(i,j).

**Yêu cầu:** Cho biết một phương án chuyển với chi phí thấp nhất.

**Dữ liệu** vào từ file ‘MESSAGES.INP’:

* Dòng đầu tiên: 2 số n và m;
* Dòng thứ i trong số n dòng tiếp theo: Dãy m số nguyên dương b1, b2, ..., bm trong đó bj là chi phí chuyển i gói tin trên kênh j.

**Kết quả** ghi vào file ‘MESSAGES.OUT’:

* Dòng đầu tiên: tổng chi phí thấp nhất theo phương án tìm được;
* Dòng thứ j trong số m dòng tiếp theo: số lượng gói tin chuyển trên kênh j.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **MESSAGES.INP** |  | **MESSAGES.OUT** |
| 5 4  31 10 1 1  1 31 12 13  4 10 31 1  6 1 20 19  10 5 10 10 |  | 2  0  4  1  0 |

***Giải thích:*** *Với n = 5 gói tin, m = 4 kênh và chi phí c(i,j) cho trước, trong đó i là chỉ số dòng (số gói tin), j là chỉ số cột (kênh) thì cách chuyển sau đây sẽ cho chi phí thấp nhất là 2*

|  |  |  |
| --- | --- | --- |
| *Kênh* | *Số gói tin* | *Chi phí* |
| 1 | 0 | 0 |
| 2 | 4 | 1 |
| 3 | 1 | 1 |
| 4 | 0 | 0 |

1. **Truyền tin ngắn nhất Tên file chương trình ‘MESS.???’**

Trong một mạng gồm n máy tính đánh số từ 1 đến n. Sơ đồ nối mạng được cho bởi hệ thống gồm m kênh nối trực tiếp giữa một số cặp máy trong mạng. Biết chi phí truyền cho 1 đơn vị thông tin theo mỗi kênh nối của mạng.

Người ta cần chuyển 1 bức thông điệp từ máy s -> t. Để đảm bảo an ninh, người ta muốn chuyển bức thông điệp này theo 2 đường truyền tin khác nhau (tức là không có kênh nào của mạng được sử dụng trong cả 2 đường truyền tin). Chi phí của 1 đường truyền được hiểu là tổng chi phí trên các kênh của nó.

**Yêu cầu:** Giả sử bức thông điệp có độ dài một đơn vị thông tin, hãy tìm cách chuyển thông điệp từ s -> t sao cho tổng chi phí chuyển thông tin ( bằng tổng chi phí theo cả hai đường truyền tin) là nhỏ nhất.

**Dữ liệu** vào từ file ‘MESS.INP’

* Dòng đầu ghi n, m, s, t cách nhau bởi dấu cách (n<=100).
* Mỗi dòng thứ i trong số m dòng tiếp theo ghi thông tin về kênh nối thứ i của mạng gồm 3 số di, ci, gi: di, ci là chỉ số 2 máy tương ứng và gi là chi phí để truyền theo kênh này (đường truyền là hai chiều).

**Kết quả** ghi vào file ‘MESS.OUT’

* Dòng đầu ghi tổng chi phí.
* Dòng thứ hai ghi đường truyền 1: s-> t.
* Dòng thứ ba ghi đường truyền 2: s ->t.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **MESS.INP** |  | **MESS.OUT** |
| 5 7 15  1 2 3  1 4 8  2 3 5  2 4 4  3 5 5  4 3 8  4 5 3 |  | 24  1 2 3 5  1 4 5 |

1. **Cắt bánh Tên file chương trình ‘CAKE.???’**

Để ghi tên vào danh sách các kỷ lục Thế giới, David và các bạn làm một chiếc bánh khổng lồ có hình một đa giác lồi. Sau khi kỷ lục đã được ghi nhận, Mọi người quyết định cắt bánh chia cho các thành viên tham gia làm bánh theo qui luật như sau: Mỗi lần chọn 1 đỉnh của đa giác rồi cắt bỏ đỉnh đó bằng cách cắt qua 2 đỉnh kề bên. Phần bánh hình tam giác có được từ mỗi lần cắt như vậy sẽ chia cho các thí sinh. Công việc cắt bánh sẽ tiếp tục cho đến khi chiếc bánh có dạng một tứ giác. Miếng bánh cuối cùng này sẽ giành cho người cắt bánh.

David được chọn làm người cắt bánh và lẽ dĩ nhiên, anh ta muốn cắt bánh sao cho phần của mình có diện tích lớn nhất có thể được..

**Yêu cầu:** Hãy tính xem phần diện tích lớn nhất của miếng bánh hình tứ giác mà David có thể có được là bao nhiêu.

**Dữ liệu:** Trong file văn bản CAKE.INP gồm

* Dòng thứ nhất ghi số N là số đỉnh của đa giác.
* N dòng tiếp theo là các cặp số nguyên biểu diễn các đỉnh của đa giác. Các đỉnh được liệt kê theo chiều kim đồng hồ.

**Kết quả:** Xuất ra file văn bản CAKE.OUT gồm 1 số duy nhất ghi diện tích lớn nhất của tứ giác tìm được. Kết quả lấy chính xác tới 1 chữ số phần thập phân.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **CAKE.INP** |  | **CAKE.OUT** |
| 6  2 1  2 3  5 7  8 3  8 1  5 0 |  | 21.0 |

***Giới hạn kỹ thuật:***

* *4 ≤ N ≤ 1500*
* *|xi|, |yi| ≤ 15000*
* *Thời gian: 1 giây*

1. **Bao lồi chuẩn Tên file chương trình ‘ENVEL.???’**

Bao lồi chuẩn của N điểm cho trước là đa giác lồi diện tích nhỏ nhất chứa các điểm đã cho (nên trong, trên cạnh hay ở đỉnh) và có các cạnh song song với trục toạ độ hoặc làm thành một góc 450 với trục toạ độ.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Cho N điểm với các toạ độ nguyên (xi, yi), 0≤xi, yi≤1000, i=1,2,...,N, 1≤N≤106. Các điểm có thể trùng nhau.

**Yêu cầu:**xác định bao lồi chuẩn của các đỉnh đã cho bằng cách chỉ ra các đỉnh của nó theo trình tự nối ngược chiều kim đồng hồ.

**Dữ liệu** vào từ file ‘ENVEL.INP’ mỗi dòng chứa hai số nguyên là toạ độ của một điểm.

**Kết quả**ghi vào file ‘ENVEL.OUT’ toạ độ của đỉnh bao lồi chuẩn theo trình tự ngược chiều kim đồng hồ, mỗi dòng chứa hai số nguyên - toạ độ của một đỉnh (Hoành độ nhỏ nhất được ghi trước, nếu có 2 đỉnh cùng hoành độ thì lấy đỉnh ở trên cao trước).

**Ví dụ :**

|  |  |  |
| --- | --- | --- |
| **ENVEL.INP** |  | **ENVEL.OUT** |
| 3 3  3 1  2 2  4 2  3 2 |  | 2 2  3 1  4 2  3 3 |

1. **Xếp hạng Tên file chương trình ‘RANK.???’**

Trong đợt tổng kết cuộc thi tin học do tỉnh Đoàn tổ chức, ban tổ chức cuộc thi cần chọn ra k (k<=2000) thí sinh có điểm số cao nhất để trao giải thưởng. Danh sách tên và điểm của n (n<=106) thí sinh dự thi được ghi trên một file văn bản, giả thiết rằng không có hai thí sinh nào có cùng số điểm.

**Yêu cầu:** đưa ra danh sách k thí sinh có điểm cao nhất của cuộc thi.

**Dữ liệu** vào từ file ‘RANK.INP’:

* Dòng đầu tiên ghi 2 số nguyên dương n, k
* Tiếp đến là n nhóm dòng, mỗi nhóm dòng gồm 2 dòng, trong đó dòng đầu ghi họ tên (là dãy gồm không quá 10 ký tự) của thí sinh dự thi, dòng thứ hai ghi điểm số mà anh ta đạt được.

**Kết quả** ghi vào file ‘RANK.OUT’ gồm k nhóm dòng tương ứng với k thí sinh có điểm cao nhất của cuộc thi theo thứ tự điểm số giảm dần.

* Dòng đầu tiên trong nhóm ghi họ tên.
* Dòng thứ hai ghi điểm số.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **RANK.INP** |  | **RANK.OUT** |
| 6 3  A  9  B  7  C  12  D  8  E  5  F  11 |  | C  12  F  11  A  9 |

1. **Thực hiện phép chia Tên file chương trình ‘DIVISION.???’**

Cho 2 số nguyên dương n và m.

**Yêu cầu:** Hãy thực hiện phép chia n cho m (kết quả biểu diễn ở hệ thập phân hữu hạn hoặc vô hạn tuần hoàn)

**Dữ liệu** vào từ file văn bản ‘DIVISION.INP’ gồm nhiều dòng: mỗi dòng một cặp số n, m (cách nhau 1 dấu cách)

**kết quả** ghi vào file văn bản ‘DIVISION.OUT’ ghi kết quả của phép chia tương ứng với cặp số ở file input (lấy 250 chữ số phần thập phân nếu có với trường hợp vô hạn không tuần hoàn).

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **DIVISION.INP** |  | **DIVISION.OUT** |
| 100 5  32567 125  11113 72  367 12  5 7  67 737 |  | 20  260.536  154.347(2)  30.58(3)  0.(714285)  0.(09) |

1. **Dãy con lớn nhất Tên file chương trình ‘ARRMAX.???’**

Trên một vòng tròn người ta đánh dấu n vị trí. Các vị trí được đánh số thứ tự từ 1 đến n theo chiều kim đồng hồ. Tại vị trí i người ta ghi số nguyên ai (i=1,2,...,n). Cần tìm cách chọn ra dãy con gồm một số số liên tiếp trên vòng tròn (theo chiều kim đồng hồ) có tổng các số hạng là lớn nhất.

**Dữ liệu** vào từ file ‘ARRMAX.INP’:

* Dòng đầu tiên ghi số nguyên dương n (n≤107);
* Dòng thứ i trong số N dòng tiếp theo ghi số ai ()

**Kết quả** ghi vào file ‘ARRMAX.OUT’ giá trị tổng các số hạng của dãy tìm được

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **ARRMAX.INP** |  | **ARRMAX.OUT** |
| 7  2  -4  1  -7  4  6  -1 |  | 11 |

1. **Danh sách vòng Tên file chương trình ‘CLIST.???’**

Để làm việc với một danh sách gồm n số nguyên cần phải có 2 thao tác. Thao tác Top chuyển phần tử đầu tiên của danh sách xuống vị trí cuối cùng của danh sách, còn thao tác Bottom chuyển phần tử cuối cùng của danh sách lên vị trí đầu tiên của danh sách. Ta gọi một phép biến đổi danh sách đã cho là việc thực hiện đầu tiên là k thao tác Top, tiếp đến là l thao tác Bottom. Do đó lần thực hiện các thao tác với danh sách là rất lớn nên đòi hỏi phải có những thủ tục thực hiện hiệu quả hai thao tác nói trên để thực hiện liên tiếp x phép biến đổi để chuyển danh sách về trạng thái cuối cùng.

**Yêu cầu:** Viết chương trình cho phép đối với một danh sách và hai số k, l cho trước, xác định trạng thái của danh sách sau x lần thực hiện phép biến đổi.

**Dữ liệu** vào từ file ‘CLIST.INP’:

* Dòng đầu tiên là 3 số nguyên dương n, k, l (1≤n, k, l≤100)
* Dòng thứ hai chứa n số nguyên, mỗi số có trị tuyệt đối không vượt quá 10000, được sắp xếp theo thứ tự tương ứng với trạng thái khởi đầu của danh sách.
* Dòng thứ ba chứa số nguyên x (0≤x≤2\*109)

**Kết quả** ghi vào file ‘CLIST.OUT’ trên một dòng các phần tử của danh sách được xếp lại sau X phép biến đổi .

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **CLIST.INP** |  | **CLIST.OUT** |
| 5 2 1  1 2 3 4 5  10 |  | 1 2 3 4 5 |