

## Năm 2005

### Bảng A

#### Bài 1. Phân đoạn

Tên file chương trình: SEGPAS.PAS

Cho dãy số nguyên  $a_1, a_2, \dots, a_n$  và số nguyên dương  $k$ . Ta gọi  $k$ -phân đoạn của dãy số đã cho là cách chia dãy số đã cho ra thành  $k$  đoạn, mỗi đoạn là một dãy con gồm các phần tử liên tiếp của dãy. Chính xác hơn, một  $k$ -phân đoạn được xác định bởi dãy chỉ số

$$1 \leq n_1 < n_2 < \dots < n_k = n.$$

Đoạn thứ  $i$  là dãy con  $a_{n_{i-1}+1}, a_{n_{i-1}+2}, \dots, a_{n_i}$ ,  $i = 1, 2, \dots, k$ . Ở đây quy ước  $n_0 = 0$ .

**Yêu cầu:** Hãy xác định số  $M$  nhỏ nhất để tồn tại  $k$ -phân đoạn sao cho tổng các phần tử trong mỗi đoạn đều không vượt quá  $M$ .

**Dữ liệu:** Vào từ file văn bản SEGPAS.INP.

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $k$  ( $1 \leq k \leq n \leq 15000$ );
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo chứa số nguyên  $a_i$  ( $|a_i| \leq 30000$ ),  $i = 1, 2, \dots, n$ .

Các số cạnh nhau trên một dòng trong file dữ liệu cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file SEGPAS.OUT một số nguyên duy nhất là giá trị  $M$  tìm được.

**Ví dụ:**

SEGPAS . INP	
9	4
1	
1	
1	
3	
2	
2	
1	
3	
1	

SEGPAS . OUT
5

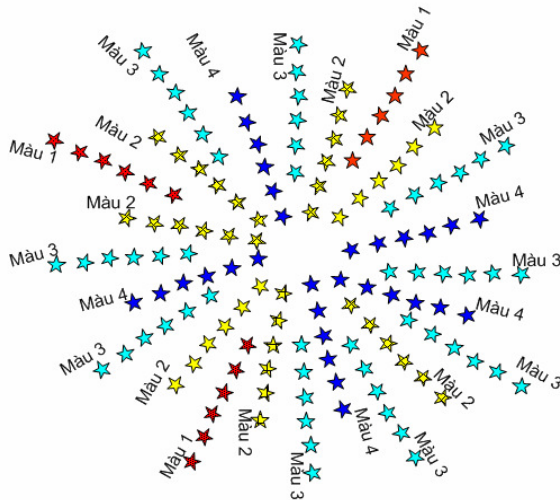
**Bài 2: Pháo hoa****Tên chương trình: FIREWK.PAS**

Nhằm chào mừng các ngày lễ lớn trong năm 2005 người ta đã chế tạo một loại đạn pháo hoa mới, khi bắn, đạn nổ thành bông hoa  $2n$  cánh màu ( $1 \leq n \leq 30$ ). Nguyên vật liệu cho phép tạo được  $m$  màu khác nhau, đánh số từ 1 đến  $m$  ( $2 \leq m \leq 32$ ).

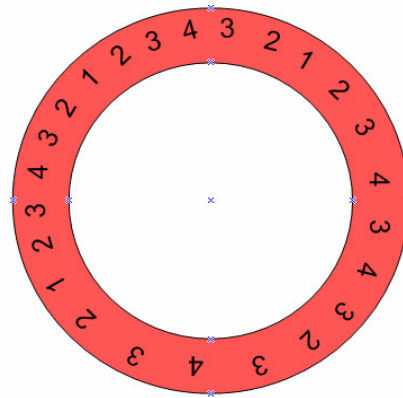
Để đảm bảo tính mỹ thuật, việc chuyển tiếp màu giữa 2 cánh hoa kề nhau phải tuân theo quy tắc chuyển màu cầu vồng sau đây:

- Bên cạnh cánh hoa màu  $i$  phải là cánh hoa màu  $i-1$  hoặc  $i+1$ , với  $1 < i < m$ ,
- Bên cạnh cánh hoa màu 1 chỉ có thể là cánh hoa màu 2,
- Bên cạnh cánh hoa màu  $m$  chỉ có thể là cánh hoa màu  $m-1$ .

Một bông hoa không nhất thiết phải có đầy đủ  $m$  màu. Mỗi bông hoa tương ứng với một vòng tròn  $2n$  số thể hiện màu của các cánh hoa. Ví dụ, hình 1 là bông hoa 24 cánh ( $n = 12$ ) và hình 2 là vòng tròn số



Hình 1



Hình 2

tương ứng với nó. Mỗi bông hoa được mô tả bằng dãy  $2n$  số nguyên liệt kê các chỉ số màu của các cánh hoa theo chiều kim đồng hồ. Ví dụ, bông hoa ở hình 1 có thể được mô tả bằng dãy số

**3 4 3 2 1 2 3 4 3 2 1 2 3 4 3 2 1 2 3 4 3 2.**

Dãy có thứ tự từ điển nhỏ nhất trong các dãy có thể dùng để mô tả hoa được gọi là mã hoa. Khi đó, mã hoa của bông hoa ở hình 1 sẽ là

**1 2 3 4 3 2 1 2 3 4 3 2 1 2 3 4 3 2 1 2 3 4 3 2.**

Trong các ngày lễ, Ban tổ chức yêu cầu bắn các đạn pháo hoa  $2n$  cánh có đúng  $k$  cánh màu  $C$  ( $0 \leq k \leq 2$ ). Các mã hoa thỏa mãn yêu cầu vừa nêu cũng được sắp xếp theo thứ tự từ điển và đánh số bắt đầu từ 1. Hơn nữa, nhằm tạo ra các hoa không giống nhau, đội bắn pháo hoa cần đảm bảo hai viên đạn pháo hoa bắn liên tiếp phải có mã khác nhau. Do vậy, người ta đã thiết kế một Hệ thống chụp ảnh và phân tích tự động để báo cho đội bắn pháo hoa biết số thứ tự của viên đạn pháo hoa vừa nổ trên trời. Em được giao viết chương trình giải quyết nhiệm vụ chính trong phần mềm phân tích tự động này.

**Yêu cầu:** Cho biết  $n, m, k$  và  $C$ . Gọi  $X$  là tập tất cả các mã hoa  $2n$  cánh có đúng  $k$  cánh màu  $C$ .

- Hãy xác định số lượng  $p$  các phần tử của  $X$ ;
- Cho một mã hoa nào đó trong tập  $X$ . Hãy xác định số thứ tự từ điển của nó trong  $X$ .

**Dữ liệu:** Vào từ file văn bản FIREWK.INP. Dòng đầu tiên chứa 4 số nguyên  $n, m, k, C$ ; dòng tiếp theo chứa  $2n$  số nguyên mô tả một mã hoa.

Các số trên một dòng của file dữ liệu cách nhau ít nhất một dấu cách.

**Kết quả:** Đưa ra file văn bản FIREWK.OUT. Dòng đầu tiên ghi số nguyên  $p$ ; dòng tiếp theo ghi số thứ tự tìm được của mã hoa.

**Ví dụ:**

FIREWK.INP			
3	4	0	1
2	3	4	3 4 3

FIREWK.OUT	
4	
3	

**Bài 3. Bộ sưu tập****Tên chương trình: COLLECT.PAS**

Một bộ sưu tập tiền xu cổ được coi là có giá trị phải gồm không ít hơn  $Z_0$  đồng tiền vàng,  $S_0$  đồng tiền bạc và  $M_0$  đồng tiền đồng. Bộ sưu tập ban đầu của Alibaba có một số lượng nhất định các đồng tiền vàng, bạc và đồng nhưng chưa phải là một bộ sưu tập có giá trị. Tại Trụ sở của Hiệp hội những người sưu tầm tiền cổ có đặt một máy đổi tiền để giúp hội viên đổi được các bộ sưu tập có giá trị. Tuy nhiên, máy đổi chỉ hỗ trợ việc đổi tiền trọn gói theo quy tắc đổi gói  $(Z_1, S_1, M_1)$  lấy gói  $(Z_2, S_2, M_2)$  đồng tiền. Các quy tắc đổi tiền khác nhau từng đôi một, được gán số hiệu tuần tự 1, 2, 3, ... và được công bố trước. Hội viên có thể tạo gói tiền thích hợp từ bộ sưu tập của mình để thực hiện việc đổi tiền. Các đồng tiền nhận được sau mỗi lần đổi được gộp lại với các đồng tiền mà hội viên đang có để thành *một bộ sưu tập mới* và có thể được sử dụng để đổi trong những lần sau nếu cần. Số lần đổi không hạn chế, tuy nhiên, là người thực dụng, *Alibaba luôn cố gắng giảm tới mức tối đa số lần đổi tiền*. Mặt khác, để ngăn chặn việc đầu cơ, Hiệp hội quy định, trong mọi thời điểm, mỗi hội viên không được giữ quá 4 đồng tiền mỗi loại và không được phép đổi tiếp khi đã đổi được một bộ sưu tập có giá trị.

**Yêu cầu:** Cho biết số lượng  $Z, S, M$  các đồng tiền vàng, bạc, đồng mà Alibaba có ban đầu và các quy tắc đổi tiền. Hãy chỉ ra tất cả các bộ sưu tập tiền cổ có giá trị mà Alibaba có thể có được sau một số lần đổi không vượt quá  $k$  cho trước.

**Dữ liệu:** Vào từ file văn bản COLLECT.INP.

- Dòng đầu ghi số nguyên dương  $k (k \leq 1000)$ ; dòng thứ 2 ghi 6 số nguyên không âm  $Z, S, M, Z_0, S_0, M_0 (0 \leq Z, S, M, Z_0, S_0, M_0 \leq 4)$ ;
- Các dòng tiếp theo mỗi dòng ghi 6 số nguyên không âm  $Z_1, S_1, M_1, Z_2, S_2, M_2$  xác định một quy tắc đổi tiền  $(0 \leq Z_1, Z_2, S_1, S_2, M_1, M_2 \leq 4)$ .

**Kết quả:** Đưa ra file văn bản COLLECT.OUT. Nếu không tồn tại cách đổi để có được bộ sưu tập có giá trị, file kết quả chỉ gồm một số -1. Trong trường hợp ngược lại, dòng đầu ghi số  $v$  là số các bộ tiền cổ có giá trị mà Alibaba có thể đổi được. Dòng thứ  $i$  trong  $v$  dòng tiếp theo ghi 4 số nguyên  $Z_i, S_i, M_i, k_i$  mô tả bộ sưu tập có giá trị thứ  $i$  và số lần đổi  $k_i$  ít nhất không vượt quá  $k$  cần thực hiện để có được bộ sưu tập ấy.

Các số trên một dòng của file dữ liệu và file kết quả đặt cách nhau ít nhất một dấu cách.

**Ví dụ:**

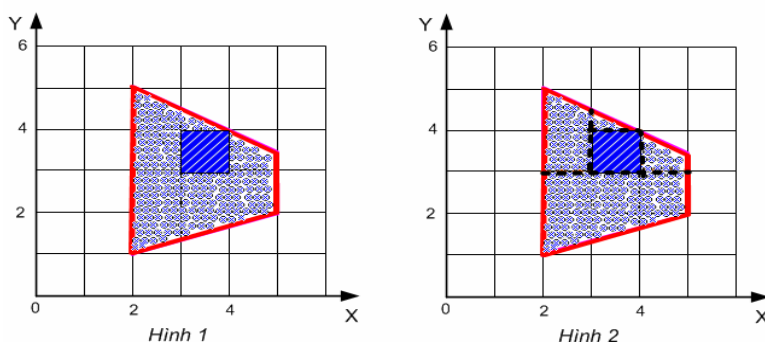
COLLECT . INP						
2						
4	0	1	3	3	3	
1	0	1	1	1	1	
2	0	1	1	3	3	

COLLECT . OUT				
1				
3	3	3	1	

**Bài 4. Khuôn thép****Tên chương trình: STEEL.PAS**

Để chuẩn bị cho Lễ hội kỷ niệm 30 năm ngày Chiến dịch Hồ Chí Minh toàn thắng, giải phóng miền Nam, thống nhất đất nước, người ta cần gia công các loại khuôn thép có hình dạng là các hình đa giác lồi  $M$  đỉnh. Mỗi khuôn thép được thiết kế trên một tấm thép cũng có hình dạng là một hình đa giác lồi  $N$  đỉnh, không có cạnh nào của khuôn thép nằm gọn trên một cạnh của tấm thép. Để tiện cho việc gia công, khuôn thép được vẽ sao cho hai đường thẳng chứa hai cạnh không kề nhau của nó không cắt nhau ở bên trong tấm thép.

Công việc chính cần làm trong quá trình gia công là sử dụng máy cắt để cắt được khuôn thép từ tấm thép ra. Rõ ràng là cần phải thực hiện  $M$  nhát cắt. Mỗi nhát cắt được thực hiện bằng cách chọn một cạnh nào đó của khuôn thép và cắt theo đường thẳng chứa cạnh ấy chia tấm thép thành hai phần, một phần chứa khuôn thép cần gia công. Chi phí cắt khuôn thép là tổng chiều dài của các đường cắt.



Trên hình 1 và 2, tấm thép là tứ giác được tô nhạt, khuôn thép là hình vuông được tô bằng các gạch đậm. Các nét gạch đứt là các đường cắt với tổng chi phí bằng 6.5 đơn vị.

**Yêu cầu:** Cho biết hình dạng tấm thép và khuôn thép cần gia công. Hãy tìm phương án cắt khuôn thép có chi phí nhỏ nhất.

**Dữ liệu:** Vào từ file văn bản STEEL.INP: Dòng đầu ghi số  $N (3 \leq N \leq 2000)$  là số đỉnh của tấm thép;  $N$  dòng tiếp theo, mỗi dòng ghi 2 số thực  $x$  và  $y$  ( $-10^4 < x, y < 10^4$ ), là tọa độ  $N$  đỉnh của tấm thép được liệt kê theo chiều kim đồng hồ bắt đầu từ một đỉnh nào đó; dòng tiếp theo ghi số  $M (3 \leq M \leq 2000)$  là số đỉnh của khuôn thép; cuối cùng là  $M$  dòng, mỗi dòng ghi 2 số thực  $x$  và  $y$  ( $-10^4 < x, y < 10^4$ ) là tọa độ  $M$  đỉnh của khuôn thép được liệt kê theo chiều kim đồng hồ bắt đầu từ một đỉnh nào đó. Các số trên một dòng cách nhau ít nhất một dấu cách.

**Kết quả:** Đưa ra file văn bản STEEL.OUT chi phí nhỏ nhất tìm được với độ chính xác tới 4 chữ số sau dấu chấm thập phân.

**Ví dụ:**

STEEL . INP	
4	
2	1
2	5
5	3.5
5	2
4	
3	3
3	4
4	4
4	3

STEEL . OUT
6.5000

## Bảng B

### Bài 1. Đóng gói sản phẩm

*Tên file chương trình: ZXY.PAS*

Ở đầu ra của một dây chuyền sản xuất trong nhà máy ZXY có một máy xếp tự động. Sau khi kết thúc việc gia công trên dây chuyền, các sản phẩm sẽ được xếp vào các hộp có cùng dung lượng  $M$ . Sản phẩm rời khỏi dây chuyền được xếp vào hộp đang mở (khi bắt đầu ca làm việc có một hộp rỗng được mở sẵn) nếu như dung lượng của hộp còn đủ để chứa sản phẩm. Trong trường hợp ngược lại, máy sẽ tự động đóng nắp hộp hiện tại, cho xuất xưởng rồi mở một hộp rỗng mới để xếp sản phẩm vào. Trong một ca làm việc có  $n$  sản phẩm đánh số từ 1 đến  $n$  theo đúng thứ tự mà chúng rời khỏi dây chuyền. Sản phẩm thứ  $i$  có trọng lượng là  $a_i$ ,  $i = 1, 2, \dots, n$ . Ban Giám đốc nhà máy qui định rằng sản phẩm xuất xưởng của mỗi ca làm việc phải được xếp vào trong không quá  $k$  hộp.

**Yêu cầu:** Hãy giúp người quản đốc của ca làm việc xác định giá trị  $M$  nhỏ nhất sao cho số hộp mà máy tự động cần sử dụng để xếp đầy  $n$  sản phẩm xuất xưởng của ca không vượt quá số  $k$  cho trước.

**Dữ liệu:** Vào từ file văn bản ZXY.INP:

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $k$ , ( $1 \leq k \leq n \leq 15000$ );
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa số nguyên dương  $a_i$  ( $a_i \leq 30000$ ),  $i = 1, 2, \dots, n$ .

Các số trên một dòng cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file ZXY.OUT một số nguyên duy nhất là dung lượng của hộp.

**Ví dụ:**

ZXY . INP	ZXY . OUT
9 4	5
1	
1	
1	
3	
2	
2	
1	
3	
1	

**Bài 2. Chữ số****Tên chương trình: DIGIT.PAS**

Cho chuỗi  $M$  không quá 127 ký tự lấy từ tập  $F = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$  và không bắt đầu bằng ký tự 0. Gọi  $S$  là chuỗi với giá trị ban đầu là chuỗi  $M$ .

Người ta biến đổi  $M$  theo quy tắc sau: đếm số lần xuất hiện các ký tự  $0, 1, 2, \dots, F$ , gọi  $K_i$  là số lần xuất hiện ký tự  $i$  (với  $i$  lần lượt là  $0, 1, 2, \dots, F$ ). Với các  $K_i \neq 0$  người ta viết liên tiếp chuỗi biểu diễn số  $K_i$  trong cơ số 16 và ký tự  $i$ . Chuỗi kết quả thu được là giá trị mới của  $M$ . Sau mỗi lần biến đổi người ta lại viết tiếp  $M$  vào sau  $S$ .

Ví dụ, với  $M = '150A'$ ,  $S$  nhận giá trị ban đầu là  $'150A'$ .

Sau lần biến đổi thứ nhất ta có  $M$  là  $'1011151A'$  và  $S = '150A1011151A'$ .

Sau lần biến đổi thứ 2 ta có  $M$  là  $'1051151A'$  và  $S = '150A1011151A1051151A'$ .

Sau lần biến đổi thứ 3 ta có  $M$  là  $'1041251A'$  và  $S = '150A1011151A1051151A1041251A'$ .

**Yêu cầu:** Cho chuỗi  $M$ , số lần biến đổi  $L$  ( $0 \leq L \leq 10^7$ ) và  $X$  là một ký tự từ tập  $F$ . Hãy đếm số lần xuất hiện  $X$  trong  $S$  thu được sau  $L$  lần biến đổi  $M$ .

**Dữ liệu:** Vào từ file văn bản DIGIT.INP :

- Dòng thứ nhất chứa chuỗi  $M$ ,
- Dòng thứ 2 chứa số nguyên  $L$
- Dòng thứ 3 chứa ký tự  $X$ .

**Kết quả:** Đưa ra file văn bản DIGIT.OUT một số nguyên - số lần xuất hiện  $X$ .

**Ví dụ:**

DIGIT . INP
150A
3
2

DIGIT . OUT
1

**Bài 3. Đổi đất****Tên file chương trình: LAND.PAS**

Hai bộ lạc Anpha và Bêta sống rất hoà thuận với nhau. Một phần ranh giới của hai bộ lạc là một đường gấp khúc không tự cắt. Đường gấp khúc nhận được bằng cách lần lượt nối  $N$  điểm đôi một khác nhau  $A_1, A_2, \dots, A_N$ . Điểm  $A_i$  được xác định bởi hoành độ  $x_i$  và tung độ  $y_i$  ( $x_i$  là các số nguyên thoả mãn điều kiện:  $x_i \leq x_{i+1}$ ). Phần đất của bộ lạc Anpha nằm ở phía trên đường gấp khúc.

Nhân dịp năm mới, tù trưởng hai bộ lạc quyết định thay đường ranh giới cũ bằng cách xây dựng một đường cao tốc nối thẳng từ  $A_1$  tới  $A_N$  và lấy đường cao tốc này ranh giới mới. Dĩ nhiên, sự thay đổi này sẽ chuyển một phần đất của bộ lạc Anpha cho bộ lạc Bêta và ngược lại. trưởng thoả thuận phần diện tích chênh lệch do việc thay đường ranh giới sẽ được điều chỉnh trong tương lai bằng cách khác.

**Yêu cầu:** Hãy tính diện tích phần đất  $SA$  của bộ lạc trở thành đất của bộ lạc Bêta và diện tích phần đất  $SB$  của bộ lạc Bêta trở thành đất của bộ lạc Anpha sau thay đổi đường ranh giới giữa hai bộ lạc.

**Dữ liệu:** Vào từ file văn bản LAND.INP trong đó :

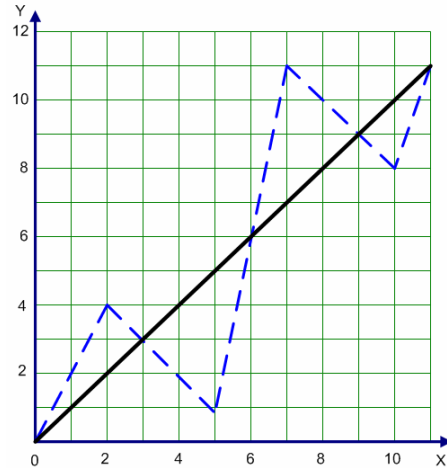
- Dòng đầu chứa số  $N$  ( $N \leq 10000$ ) ;
- Dòng thứ  $i$  trong  $N$  dòng tiếp theo chứa hai số nguyên  $x_i$  và  $y_i$  đặt cách nhau ít nhất 1 dấu cách ( $-32000 \leq x_i, y_i \leq 32000$ ).

**Kết quả:** Đưa ra file văn bản LAND.OUT trong đó dòng thứ nhất chứa  $SA$ , dòng thứ hai chứa  $SB$ . Kết quả được lấy chính xác với 4 chữ số sau dấu chấm thập phân.

**Ví dụ :**

LAND . INP	
6	
0	0
2	4
5	1
7	11
10	8
11	11

LAND . OUT	
8.0000	
9.0000	



đổi  
tốc là  
làm  
số  
Hai từ  
đổi  
một  
  
Anpha  
khi

**Bài 4. Bộ sưu tập****Tên chương trình: COLLECT.PAS**

Một bộ sưu tập tiền xu cổ được coi là có giá trị phải gồm không ít hơn  $Z_0$  đồng tiền vàng,  $S_0$  đồng tiền bạc và  $M_0$  đồng tiền đồng. Bộ sưu tập ban đầu của Alibaba có một số lượng nhất định các đồng tiền vàng, bạc và đồng nhưng chưa phải là một bộ sưu tập có giá trị. Tại Trụ sở của Hiệp hội những người sưu tầm tiền cổ có đặt một máy đổi tiền để giúp hội viên đổi được các bộ sưu tập có giá trị. Tuy nhiên, máy đổi chỉ hỗ trợ việc đổi tiền trọn gói theo quy tắc đổi gói  $(Z_1, S_1, M_1)$  lấy gói  $(Z_2, S_2, M_2)$  đồng tiền. Các quy tắc đổi tiền khác nhau từng đôi một, được gán số hiệu tuần tự 1, 2, 3, ... và được công bố trước. Hội viên có thể tạo gói tiền thích hợp từ bộ sưu tập của mình để thực hiện việc đổi tiền. Số lần đổi tiền là không hạn chế, tuy nhiên, để ngăn chặn việc đầu cơ, Hiệp hội quy định mỗi hội viên không được giữ quá 4 đồng tiền mỗi loại. Các đồng tiền nhận được sau mỗi lần đổi được gộp lại với các đồng tiền mà hội viên đang có để thành *một bộ sưu tập mới* và có thể được sử dụng để đổi trong những lần sau nếu cần.

**Yêu cầu:** Cho biết số lượng  $Z, S, M$  các đồng tiền vàng, bạc, đồng mà Alibaba có ban đầu và các quy tắc đổi tiền. Hãy chỉ ra *một phương án đổi tiền* nào đó để Alibaba có được bộ sưu tập có giá trị. Dữ liệu vào đảm bảo luôn có phương án.

**Dữ liệu:** Vào từ file văn bản COLLECT.INP:

- Dòng đầu ghi 6 số nguyên không âm  $Z, S, M, Z_0, S_0, M_0$  ( $0 \leq Z, S, M, Z_0, S_0, M_0 \leq 4$ );
- Các dòng tiếp theo mỗi dòng ghi 6 số nguyên không âm  $Z_1 S_1 M_1 Z_2 S_2 M_2$  xác định một quy tắc đổi tiền.

**Kết quả:** Đưa ra file văn bản COLLECT.OUT một dòng ghi dãy số hiệu các quy tắc theo thứ tự đã sử dụng trong phương án đổi tiền.

Các số trên một dòng của file dữ liệu và file kết quả đặt cách nhau ít nhất một dấu cách.

**Ví dụ:**

COLLECT . INP					
4	0	1	3	3	3
1	0	1	0	2	2
0	1	1	0	0	3
2	0	1	1	2	3
1	0	0	1	1	0

COLLECT . OUT	
3	4



## Năm 2005

### Bài 1. Lập lịch

Tại thời điểm 0, cần phải sắp xếp thực hiện  $n$  chương trình (đánh số từ 1 đến  $n$ ) trên máy tính. Chương trình  $i$  có thời hạn hoàn thành là  $d_i$  và đòi hỏi thời gian thực hiện là  $p_i$  ( $d_i$  và  $p_i$  là các số nguyên dương). Việc thực hiện chương trình trên máy tính cần được tiến hành liên tục từ thời điểm bắt đầu đến đến thời điểm kết thúc. Khoảng thời gian thực hiện hai chương trình bất kỳ không được có nhiều hơn một điểm chung (nghĩa là hai khoảng không được giao nhau, ngoại trừ tình huống thời điểm cuối của khoảng này là trùng với thời điểm đầu của khoảng kia). Giả sử chương trình  $i$  bắt đầu thực hiện từ thời điểm  $s_i$ , khi đó  $f_i = s_i + p_i$  được gọi là thời điểm hoàn thành của nó. Ta nói chương trình  $i$  là hoàn thành đúng hạn, nếu thời điểm hoàn thành của nó là không lớn hơn  $d_i$ .

**Yêu cầu:** Hãy tìm trình tự thực hiện các chương trình sao cho có nhiều chương trình được hoàn thành đúng hạn nhất.

**Dữ liệu:** Vào từ file văn bản SCHEDULE.INP:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $n \leq 1000$ );
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo chứa thông tin về chương trình thứ  $i$  gồm hai số  $p_i, d_i$  được ghi cách nhau bởi dấu cách ( $p_i \leq d_i \leq 20000$ ).

Các số trên cùng dòng được ghi cách nhau bởi ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản SCHEDULE.OUT số lượng chương trình được hoàn thành đúng hạn theo trình tự tìm được.

**Ví dụ:**

SCHEDULE . INP
3
3 3
2 4
2 4

SCHEDULE . OUT
2

## Bài 2. Mạng đen trắng

Một hệ thống gồm  $n$  máy tính (được đánh số từ 1 đến  $n$ ) được nối với nhau thành mạng bởi các kênh nối (cho phép truyền tin hai chiều) giữa một số cặp máy tính. Mạng đã cho là *thông suốt*, nghĩa là các kênh nối đảm bảo hai máy tính bất kỳ có thể truyền tin với nhau qua kênh nối trực tiếp giữa chúng hoặc thông qua một số máy trung gian. Các kênh nối giữa các cặp máy chỉ thuộc một trong hai loại trắng hoặc đen mà để cho tiện ta sẽ gọi là kênh trắng hoặc kênh đen. Ta gọi một **phép rút gọn** mạng đã cho là một cách loại bỏ khỏi mạng một số nhiều nhất các kênh nối sao cho mạng còn lại vẫn là thông suốt.

**Yêu cầu:** Trong số các phép rút gọn mạng đã cho hãy tìm cách rút gọn sao cho mạng còn lại có chênh lệch giữa số lượng kênh trắng và số lượng kênh đen (trị tuyệt đối của hiệu giữa số lượng kênh trắng và kênh đen) là nhỏ nhất. (Nếu có nhiều lời giải thì chỉ cần đưa ra một lời giải tùy ý.)

**Dữ liệu:** Vào từ file văn bản BWNET.INP:

- Dòng đầu tiên chứa 2 số nguyên dương  $n$  và  $m$ , trong đó  $n$  là số lượng máy tính còn  $m$  là số lượng kênh nối trong mạng đã cho ( $2 < n \leq 200$ ).
- Mỗi dòng trong số  $m$  dòng tiếp theo chứa thông tin về một kênh nối bao gồm 3 số nguyên dương  $u, v, k$ , cho biết hai máy  $u, v$  được nối với nhau bởi kênh nối loại  $k$  ( $k=1$  nếu là kênh đen và  $k=2$  nếu là kênh trắng).

Các số trên cùng dòng được ghi cách nhau bởi ít nhất một dấu cách.

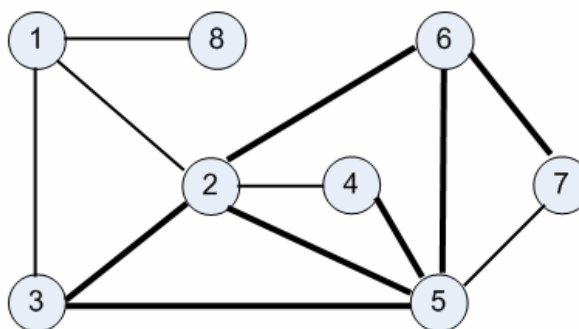
**Kết quả:** Ghi ra file văn bản BWNET.OUT:

- Dòng đầu tiên ghi hai số nguyên  $p, q$ , trong đó  $p$  là số kênh nối còn  $q$  là chênh lệch giữa số lượng kênh trắng và kênh đen trong mạng tìm được.
- $p$  dòng tiếp theo mô tả  $p$  kênh nối của mạng tìm được theo qui cách giống như trong file dữ liệu.

**Ví dụ:**

BWNET . INP	BWNET . OUT
8 12	7 1
1 8 1	3 1 1
1 2 1	4 5 2
1 3 1	5 3 2
2 3 2	6 5 2
2 4 1	7 6 2
2 5 2	2 1 1
2 6 2	8 1 1
3 5 2	
4 5 2	
5 6 2	
5 7 1	
6 7 2	

Sơ đồ nối mạng



### Bài 3. Biến đổi

Cho xâu ký tự  $S$  độ dài  $L$  ( $L \leq 20000$ ) chỉ bao gồm các chữ cái la tinh in hoa và các chữ số thập phân. Ta gọi *đoạn chữ cái* là dãy liên tiếp các ký tự chữ cái  $S_M, S_{M+1}, \dots, S_{M+P-1}$  trong xâu  $S$  thỏa mãn điều kiện: ký tự đứng trước  $S_M$  (nếu có) và ký tự đứng sau  $S_{M+P-1}$  (nếu có) phải là chữ số. Tương tự như vậy, *đoạn chữ số* là dãy liên tiếp các ký tự chữ số  $S_N, S_{N+1}, \dots, S_{N+Q-1}$  trong xâu  $S$  thỏa mãn điều kiện: ký tự đứng trước  $S_N$  (nếu có) và ký tự đứng sau  $S_{N+Q-1}$  (nếu có) phải là chữ cái.

Có 6 phép biến đổi tác động lên  $S$  (như là ví dụ lấy  $S = \text{'ABC1234EF5'}$ ):

1. Dịch chuyển vòng tròn các ký tự của  $S$  theo chiều kim đồng hồ: Ký tự  $S_L$  trở thành ký tự đầu của xâu:  $\text{ABC1234EF5} \rightarrow \text{5ABC1234EF}$ .
2. Dịch chuyển vòng tròn các ký tự của  $S$  theo chiều ngược kim đồng hồ: Ký tự  $S_1$  trở thành ký tự cuối xâu:  $\text{ABC1234EF5} \rightarrow \text{BC1234EF5A}$ ,
3. Trong một đoạn chữ số tùy chọn thay mỗi ký tự số bằng ký tự số kế tiếp theo qui tắc: Ký tự 0 - bằng ký tự 1, Ký tự 1 - bằng ký tự 2,  $\dots$ , Ký tự 9 - bằng ký tự 0,
4. Trong một đoạn chữ số tùy chọn thay mỗi ký tự số bằng ký tự số đi trước theo qui tắc: Ký tự 1 - bằng ký tự 0, Ký tự 2 - bằng ký tự 1,  $\dots$ , Ký tự 0 - bằng ký tự 9,
5. Trong một đoạn chữ cái tùy chọn thay mỗi ký tự chữ cái bằng ký tự chữ cái đi sau theo qui tắc: Ký tự A - bằng ký tự B, Ký tự B - bằng ký tự C,  $\dots$ , Ký tự Z - bằng ký tự A,
6. Trong một đoạn chữ cái tùy chọn thay mỗi ký tự chữ cái bằng ký tự chữ cái đi trước theo qui tắc: Ký tự B - bằng ký tự A, Ký tự C - bằng ký tự B,  $\dots$ , Ký tự A - bằng ký tự Z.

**Yêu cầu:** Cho hai xâu  $S1$  và  $S2$  cùng độ dài  $L$ , trong đó  $S2$  là kết quả biến đổi  $S1$  bằng cách áp dụng một số lần các phép biến đổi nêu trên. Hãy xác định số lượng tối thiểu lần biến đổi để nhận được  $S2$  từ  $S1$ .

**Dữ liệu:** Vào từ file văn bản `CHANGE.INP` gồm 2 dòng: dòng thứ nhất chứa xâu  $S1$ , dòng thứ 2 chứa xâu  $S2$ .

**Kết quả:** Đưa ra file văn bản `CHANGE.OUT` số lượng tối thiểu lần biến đổi để nhận được  $S2$  từ  $S1$

**Ví dụ:**

CHANGE . INP
ABC1234EF5
BC1234EF6A

CHANGE . OUT
2

## Bài 4. Hái nấm

Một cô bé đi hái nấm trong  $N$  khu rừng đánh số từ 1 đến  $N$ . Khu rừng thứ  $i$  có  $S_i$  cây nấm. Việc di chuyển từ khu rừng thứ  $i$  sang khu rừng thứ  $j$  tốn  $t_{ij}$  đơn vị thời gian. Đến mỗi khu rừng, cô bé có thể dừng lại để hái nấm. Nếu tổng số đơn vị thời gian cô bé dừng lại ở khu rừng thứ  $i$  là  $d_i$  ( $d_i > 0$ ), thì cô bé hái được:

$$\left\lfloor \frac{S_i}{2} \right\rfloor + \left\lfloor \frac{S_i}{4} \right\rfloor + \dots + \left\lfloor \frac{S_i}{2^{d_i}} \right\rfloor$$

cây nấm tại khu rừng đó ( $\lfloor x \rfloor$  là phần nguyên của  $x$ ). Giả thiết rằng ban đầu cô bé ở khu rừng thứ nhất và đi hái nấm trong thời gian không quá  $P$  đơn vị.

**Yêu cầu:** Hãy tính số lượng cây nấm nhiều nhất mà cô bé có thể hái được.

**Dữ liệu:** Vào từ file văn bản HAINAM.INP:

- Dòng đầu tiên chứa số nguyên dương  $N$  ( $N \leq 100$ ) và  $P$  ( $P \leq 1000$ );
- Dòng tiếp theo chứa  $N$  số nguyên dương  $S_i$  ( $S_i \leq 10000$ );
- Dòng thứ  $i$  trong  $N$  dòng cuối cùng chứa  $N$  số nguyên dương  $t_{ij}$  ( $t_{ij} \leq 1000$ ), ( $i, j = 1, \dots, N$ ).

Các số trên cùng dòng được ghi cách nhau bởi ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản HAINAM.OUT số lượng cây nấm nhiều nhất cô bé có thể hái được.

**Ví dụ:**

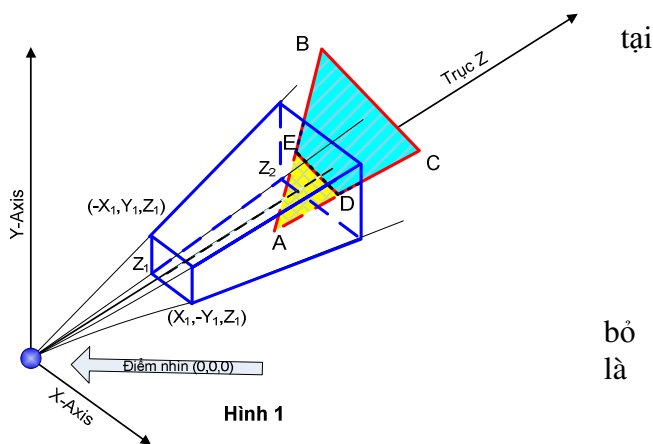
HAINAM. INP	HAINAM. OUT
2 2	3
5 1000	
0 3	
3 0	

## Bài 5. Đồ họa 3D

Trong lĩnh vực đồ họa máy tính, các hình khối được biểu diễn qua các đa giác. Trước khi hiển thị các đa giác ra card màn hình, cần phải thực hiện một số bước chuẩn bị. Hai trong những bước chuẩn bị quan trọng là: loại bỏ phần nằm ngoài vùng nhìn và tìm ra thứ tự của các hình khối theo vị trí tương đối với điểm nhìn.

“Vùng nhìn” là một hình chóp cắt có đỉnh  $(0,0,0)$ ; đáy trên là hình chữ nhật trên mặt phẳng  $z=Z_1$ , với đỉnh trái trên tại  $(-X_1, Y_1, Z_1)$ , đỉnh phải dưới tại  $(X_1, -Y_1, Z_1)$ ; đáy dưới là hình chữ nhật trên mặt phẳng  $z=Z_2$ .

Với mỗi đa giác cần vẽ, các phần của đa giác nằm ngoài “vùng nhìn” sẽ bị loại bỏ để tạo thành một đa giác mới nằm hoàn toàn trong “vùng nhìn”. Ví dụ ở hình 1, tam giác ABC sau khi loại phần BCDE nằm ngoài “vùng nhìn”, phần còn lại tam giác AED.

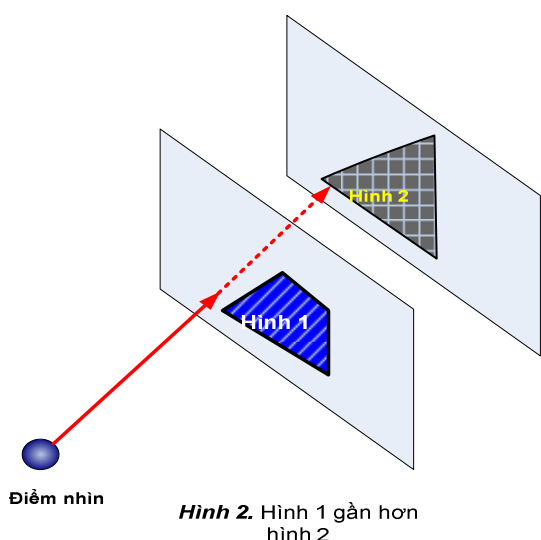


Hình 1

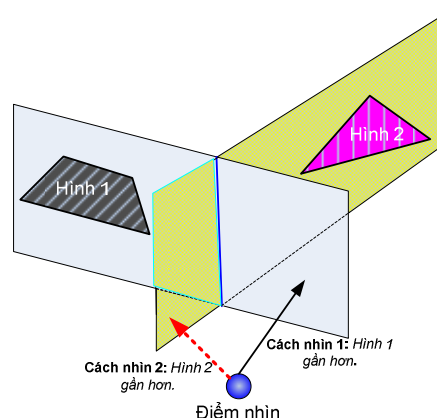
Sau khi các đa giác đã được cắt bỏ phần nằm ngoài “vùng nhìn”, chúng phải được sắp xếp lại theo vị trí tương đối với điểm nhìn. Điểm nhìn được đặt tại vị trí  $(0,0,0)$ . Để xác định thứ tự (gần, xa) của hai đa giác  $ĐG_1$  và  $ĐG_2$ , người ta kiểm tra các điều kiện:

- **ĐK1:** điểm nhìn và đa giác  $ĐG_2$  nằm ở hai nửa không gian khác nhau ngăn cách bởi mặt phẳng chứa đa giác  $ĐG_1$  và không có điểm chung với mặt phẳng chứa  $ĐG_1$ ; hoặc điểm nhìn và đa giác  $ĐG_1$  nằm ở cùng nửa không gian xác định bởi mặt phẳng chứa đa giác  $ĐG_2$  và không có điểm chung với mặt phẳng chứa  $ĐG_2$ .
- **ĐK2:** điểm nhìn và đa giác  $ĐG_1$  nằm ở hai nửa không gian khác nhau ngăn cách bởi mặt phẳng chứa đa giác  $ĐG_2$  và không có điểm chung với mặt phẳng chứa  $ĐG_2$ ; hoặc điểm nhìn và đa giác  $ĐG_2$  nằm ở cùng nửa không gian xác định bởi mặt phẳng chứa đa giác  $ĐG_1$  và không có điểm chung với mặt phẳng chứa  $ĐG_1$ .

Nếu chỉ **ĐK1** đúng (đa giác  $ĐG_1$  nằm gần điểm nhìn hơn đa giác  $ĐG_2$ ), hoặc chỉ **ĐK2** đúng (đa giác  $ĐG_2$  nằm gần điểm nhìn hơn đa giác  $ĐG_1$ ) thì thứ tự của hai đa giác  $ĐG_1$  và  $ĐG_2$  có thể xác định được (xem ví dụ ở hình 2). Nếu cả **ĐK1** và **ĐK2** cùng đúng hoặc cùng sai, thứ tự của hai đa giác  $ĐG_1$  và  $ĐG_2$  không thể xác định được (xem ví dụ ở hình 3).



Hình 2. Hình 1 gần hơn hình 2



Hình 3. Không sắp xếp được

**Yêu cầu:** Cho  $N$  hình tam giác đánh số từ 1 đến  $N$ . Hãy loại bỏ các phần nằm ngoài “vùng nhìn” của tam giác này và sau đó sắp xếp lại theo thứ tự ở từ gần đến xa đối với điểm nhìn.

**Dữ liệu:** Vào từ file văn bản DOHOA.INP:

- Dòng đầu tiên chứa số nguyên dương  $N$  ( $N \leq 100$ );
- Dòng tiếp theo chứa 4 số nguyên dương  $X_1, Y_1, Z_1, Z_2$  ( $X_1, Y_1, Z_1, Z_2 \leq 10000$ );

- Dòng thứ  $i$  trong số  $N$  dòng tiếp theo chứa chín số nguyên là tọa độ của ba đỉnh của tam giác thứ  $i$  ( $i=1, \dots, N$ ). Các tọa độ có trị tuyệt đối không vượt quá 10000.

Các số trên cùng dòng được ghi cách nhau bởi ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản DOHOA.OUT:

- Nếu có thể xác định được thứ tự giữa mọi cặp hai đa giác được tạo ra từ các tam giác ban đầu sau khi loại bỏ phần nằm ngoài vùng nhìn và có thể sắp xếp được các đa giác này theo định nghĩa thứ tự đó, hãy ghi ra chỉ số của các đa giác, theo thứ tự từ gần đến xa đối với điểm nhìn.
- Nếu không, chỉ ghi ra số -1.

Các số được ghi trên cùng một dòng, cách nhau bởi dấu cách.

**Ví dụ:**

DOHOA . INP								
2								
1	1	1	4					
1	1	2	1	3	2	4	4	2
1	1	3	1	3	3	3	3	3

DOHOA . OUT	
1	2

**Ghi chú:** Một dạng phương trình mặt phẳng:  $Ax+By+Cz+D=0$ . Một dạng phương trình đường thẳng:  $x=A_1t+A_2$ ,  $y=B_1t+B_2$ ,  $z=C_1t+C_2$  trong đó  $t$  là một tham số.  $(x,y,z)$  là tọa độ các điểm trên mặt phẳng/đường thẳng đó,  $A, B, C, D, A_1, A_2, B_1, B_2, C_1, C_2$  là các hằng số.

## Bài 6. Chia nhóm

Các trường tổ chức cho  $N$  học sinh đi cắm trại. Cần chia các học sinh này thành các nhóm không nhất thiết phải có số lượng như nhau, mỗi nhóm một trại. Mỗi nhóm phải có ít nhất một người. Số lượng nhóm và số học sinh trong mỗi nhóm sẽ được thông báo cho nhà tài trợ. Dựa trên số liệu này, nhà tài trợ sẽ tính toán số lượng *tối thiểu* chai nước uống để cấp phát sao cho: Mỗi trại đều nhận được số lượng chai nước như nhau; Số lượng các chai nước đó có thể chia đều cho các thành viên của mỗi trại. Các trường muốn tìm cách chia nhóm sao cho mỗi nhóm nhận được nhiều chai nước nhất.

**Ví dụ:** Với  $N=5$ , có 7 cách chia nhóm như sau:

- chia thành năm nhóm: mỗi nhóm có 1 người. Mỗi nhóm nhận được 1 chai nước;
- chia thành bốn nhóm: ba nhóm mỗi nhóm có 1 người, một nhóm có 2 người. Mỗi nhóm nhận được 2 chai nước;
- chia thành ba nhóm: hai nhóm mỗi nhóm có 1 người, một nhóm có 3 người. Mỗi nhóm nhận được 3 chai nước;
- chia thành ba nhóm: một nhóm có 1 người, hai nhóm mỗi nhóm có 2 người. Mỗi nhóm nhận được 2 chai nước;
- chia thành hai nhóm: một nhóm có 1 người, một nhóm có 4 người. Mỗi nhóm nhận được 4 chai nước;
- chia thành hai nhóm: một nhóm có 2 người, một nhóm có 3 người. Mỗi nhóm nhận được 6 chai nước;
- chia thành một nhóm có 5 người. Mỗi nhóm nhận được 5 chai nước.

Vậy 6 là số lượng chai nước nhiều nhất mà mỗi nhóm có thể nhận được.

**Yêu cầu:** Hãy xác định số lượng chai nước nhiều nhất mà mỗi nhóm học sinh có thể nhận được.

**Dữ liệu:** Cho 10 giá trị của  $N$ : 50, 500, 1002, 2003, 3005, 4119, 5555, 9050, 15000, 50000.

**Kết quả:** Ghi ra file văn bản CHIANHOM.OUT gồm 10 dòng, mỗi dòng chứa số lượng chai nước nhiều nhất tìm được tương ứng với một giá trị của  $N$  theo trình tự liệt kê trên.

**Lưu ý:** Thí sinh không phải nộp file chương trình mà chỉ nộp file **CHIANHOM.OUT**