Năm 2006
Bảng A

Bài 1. Dãy con dài nhất

Cho dãy số nguyên

$$a_1, a_2, ..., a_n.$$

Dãy số

$$a_i, a_{i+1}, ..., a_j$$

với $1 \le i \le j \le n$ được gọi là dãy con của dãy số đã cho và khi đó, j-i+1 được gọi là ***độ dài,*** còn $\sum_{k=i}^{j} a_k$ được gọi là ***trọng lượng*** của dãy con này.

Yêu cầu: Cho số nguyên p, trong số các dãy con của dãy số đã cho có trọng lượng không nhỏ hơn p hãy tìm dãy con có độ dài lớn nhất.

Dữ liệu: Vào từ file văn bản MAXSEQ.INP:

- Dòng đầu tiên ghi hai số nguyên n và p cách nhau bởi dấu cách;
- Dòng thứ i trong số n dòng tiếp theo chứa số nguyên a_i là số hạng thứ i của dãy số đã cho, i = 1, 2, ..., n.

Kết quả: Ghi ra file văn bản MAXSEQ.OUT số nguyên k là độ dài của dãy con tìm được (qui ước: nếu không có dãy con nào thoả mãn điều kiện đặt ra thì k = -1) .

Ví dụ:

MAXSEQ.INP	MAXSEQ.OUT
5 6 -2 3 2 -2 3	4

MAXSEQ.INP	MAXSEQ.OUT
4 9 2 3 2 -2	-1

Hạn chế: Trong tất cả các test: $1 \le n \le 20000$; $|a_i| \le 20000$; $|p| \le 10^9$. Có 50% số lượng test với $n \le 1000$.

Bài 2. Đường đi trên lưới

Cho một lưới ô vuông gồm m dòng và n cột. Các dòng được đánh số từ 1 đến m từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái qua phải. Ô nằm ở vị trí dòng i và cột j của lưới được gọi là ô (i,j) và khi đó, i được gọi là toạ độ dòng còn j được gọi là toạ độ cột của ô này. Trên ô (i,j) của lưới ghi số nguyên dương a_{ij}, $i = 1, 2, ..., m$; $j = 1, 2, ..., n$. Trên lưới đã cho, từ ô (i,j) ta có thể di chuyển đến ô (p,q) nếu các điều kiện sau đây được thoả mãn:

- $j < n$; $i \le p; j \le q$ và $i + j < p + q$;
- a_{ij} và a_{pq} có ước số chung lớn hơn 1.

Ta gọi một cách di chuyển từ mép trái sang mép phải của lưới là cách di chuyển bắt đầu từ một ô có toạ độ cột bằng 1 qua các ô của lưới tuân theo qui tắc di chuyển đã nêu và kết thúc ở một ô có toạ độ cột bằng n.

Yêu cầu: Tính số cách di chuyển từ mép trái sang mép phải của lưới.

Dữ liệu: Vào từ file văn bản NETPATH.INP:

- Dòng đầu tiên ghi 2 số nguyên dương m, n.
- Dòng thứ i trong số m dòng tiếp theo ghi n số nguyên dương $a_{i1}, a_{i2}, ..., a_{in}$ là các số trên dòng thứ i của lưới, $i = 1, 2, ..., m$.

  Hai số liên tiếp trên cùng một dòng được ghi cách bởi ít nhất một dấu cách.

Kết quả: Ghi ra file văn bản NETPATH.OUT số nguyên k là số lượng cách di chuyển tìm được, biết rằng dữ liệu đảm bảo $k < 10^9$.

Ví dụ:

NETPATH.INP	NETPATH.OUT
2 2 2 4 6 8	4

NETPATH.INP	NETPATH.OUT
2 2 2 5 6 7	0

Hạn chế: Trong tất cả các test: $1 < m, n \le 100$; $a_{ij} \le 30000$, i=1,2,...,m; j=1,2,...,n. Có 50% số lượng test với $m, n \le 50$.

Bài 3. Mạng máy tính

Một hệ thống n máy tính (các máy tính được đánh số từ 1 đến n) được nối lại thành một mạng bởi m kênh nối, mỗi kênh nối hai máy nào đó và cho phép truyền tin một chiều từ máy này đến máy kia. Giả sử s và t là hai máy tính trong mạng. Ta gọi đường truyền tin từ máy s đến máy t là một dãy các máy tính và các kênh nối chúng có dạng:

$$s = u_1, e_1, u_2, ..., u_i, e_i, u_{i+1}, ..., u_{k-1}, e_{k-1}, u_k = t,$$

trong đó $u_1, u_2, ..., u_k$ là các máy tính trong mạng, e_i – kênh truyền tin từ máy u_i đến máy u_{i+1} ($i = 1, 2, ..., k$-1).

Mạng máy tính được gọi là ***thông suốt*** nếu như đối với hai máy u, v bất kỳ ta luôn có đường truyền tin từ u đến v và đường truyền tin từ v đến u. Mạng máy tính được gọi là ***hầu như thông suốt*** nếu như đối với hai máy u, v bất kỳ, hoặc là có đường truyền tin từ u đến v hoặc là có đường truyền tin từ v đến u.

Biết rằng mạng máy tính đã cho là hầu như thông suốt nhưng không là thông suốt.

Yêu cầu: Hãy xác định xem có thể bổ sung đúng một kênh truyền tin để biến mạng đã cho trở thành thông suốt được hay không?

Dữ liệu: Vào từ file văn bản ONEARC.INP:

- Dòng đầu tiên chứa 2 số nguyên dương n và m.
- Dòng thứ i trong số m dòng tiếp theo mô tả kênh nối thứ i bao gồm hai số nguyên dương u_i, v_i cho biết kênh nối thứ i cho phép truyền tin từ máy u_i đến máy v_i, i=1,2,...,m.

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

Kết quả: Ghi ra file văn bản ONEARC.OUT:

- Dòng đầu tiên ghi 'YES' nếu câu trả lời là khẳng định, ghi 'NO' nếu câu trả lời là phủ định.
- Nếu câu trả lời là khẳng định thì dòng thứ hai ghi hai số nguyên dương u, v cách nhau bởi dấu cách cho biết cần bổ sung kênh truyền tin từ máy u đến máy v để biến mạng thành thông suốt.

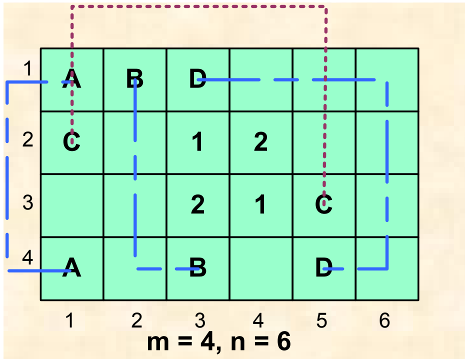
Ví dụ:

ONEARC.INP	ONEARC.OUT
3 2 1 2 2 3	YES 3 1

Hạn chế: Trong tất cả các test: $n \le 2000$, $m \le 30000$.Có 50% số lượng test với $n \le 200$.

Bài 4. Xoá cặp ô

Cho một bảng hình chữ nhật kích thước $m \times n$ ô vuông kích thước đơn vị. Các dòng được đánh số từ 1 đến m, từ trên xuống dưới. Các cột được đánh số từ 1 đến n, từ trái qua phải. Ô nằm ở vị trí dòng i và cột j của bảng được gọi là ô (i,j). Mỗi ô của bảng hoặc được để trống hoặc chứa một ký tự lấy từ tập Σ gồm các chữ số từ 0 đến 9 và các chữ cái la tinh in hoa từ A đến Z. Mỗi ký tự của tập Σ xuất hiện ở không quá 4 ô trong bảng. Hai ô chứa cùng một ký tự được gọi là giống nhau. Hai ô giống nhau có thể xoá được nếu chúng có cạnh chung hoặc tâm (giao điểm của hai đường chéo) của 2 ô này có thể nối với nhau bằng một đường gấp khúc gồm không quá 3 đoạn thẳng độ dài nguyên, mỗi đoạn song song với cạnh của bảng, và ngoại trừ hai ô cần xoá, đường gấp khúc này chỉ qua các ô trống hay nằm ngoài bảng. Các ô bị xoá trở thành ô trống. Mỗi lần xoá một cặp ô của bảng được gọi là một bước. Hình bên nêu ví dụ với trường hợp m = 4 và n = 6. Bước đầu tiên có thể xoá hai ô chứa ký tự '*A*' hoặc 2 ô chứa ký tự '*B*' hay 2 ô chứa ký tự '*D*'. Hai ô chứa ký tự '*C*' chỉ có thể xoá sớm nhất ở bước thứ 2, sau khi đã xoá các ô chứa '*A*'. Như vậy, để xoá trống 2 ô (2, 1) và (1,2) cần thực hiện tối thiểu 3 bước xoá.



Yêu cầu: Cho hai số m, n và m xâu độ dài n mô tả các dòng của bảng và hai ô khác trống (r_1, c_1), (r_2, c_2). Hãy xác định số bước ít nhất cần thực hiện để biến đổi các ô (r_1, c_1) và (r_2, c_2) trở thành ô trống.

Dữ liệu: Vào từ file văn bản DEL.INP:

- Dòng đầu tiên chứa 6 số nguyên m, n, r_1, c_1, r_2, c_2, hai số liên tiếp được ghi cách nhau bởi dấu cách.
- Dòng thứ i +1 chứa xâu n ký tự mô tả dòng thứ i của bảng (i = 1, 2, ..., m). Các ô trống được thể hiện bằng dấu chấm ('.').

Kết quả: Đưa ra file văn bản DEL.OUT số nguyên k là số bước ít nhất tìm được (qui ước: nếu không tồn tại cách biến đổi thoả mãn yêu cầu đặt ra thì k=-1).

Ví dụ:

DEL.INP	DEL.OUT
4 5 2 1 1 2 ABD... C.12.. ..21C. A.B.D.	3

DEL.INP	DEL.OUT
4 6 4 2 4 6 ABCDUV BADCVU ABCDUV BADCVU	3

Hạn chế: Trong tất cả các test: $0 < m \le 10$, $0 < n \le 20$. Có 60% số lượng test có $m \le 8$, $n \le 10$ và số lượng các ô khác trống không quá $\frac{m \times n}{2}$.

Bảng B

Bài 1. Chọn ô

Cho một bảng hình chữ nhật kích thước $4 \times n$ ô vuông. Các dòng được đánh số từ 1 đến 4, từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái qua phải. Ô nằm trên giao của dòng i và cột j được gọi là ô (i,j). Trên mỗi ô (i,j) có ghi một số nguyên a_{ij} , i =1, 2, 3, 4; j =1, 2, ..., n. Một cách chọn ô là việc xác định một tập con khác rỗng S của tập tất cả các ô của bảng sao cho không có hai ô nào trong S có chung cạnh. Các ô trong tập S được gọi là ô được chọn, tổng các số trong các ô được chọn được gọi là trọng lượng của cách chọn.

Ví dụ: Xét bảng với n=3 trong hình vẽ dưới đây

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

Cách chọn cần tìm là tập các ô S = {(3,1), (1,2), (4,2), (3,3)} với trọng lượng 32.

Yêu cầu: Hãy tìm cách chọn ô với trọng lượng lớn nhất.

Dữ liệu: Vào từ file văn bản SELECT.INP:

- Dòng đầu tiên chứa số nguyên dương n là số cột của bảng.
- Dòng thứ j trong số n dòng tiếp theo chứa 4 số nguyên a_{1j}, a_{2j}, a_{3j}, a_{4j}, hai số liên tiếp cách nhau ít nhất một dấu cách, là 4 số trên cột j của bảng.

Kết quả: Ghi ra file văn bản SELECT.OUT trọng lượng của cách chọn tìm được.

Ví dụ:

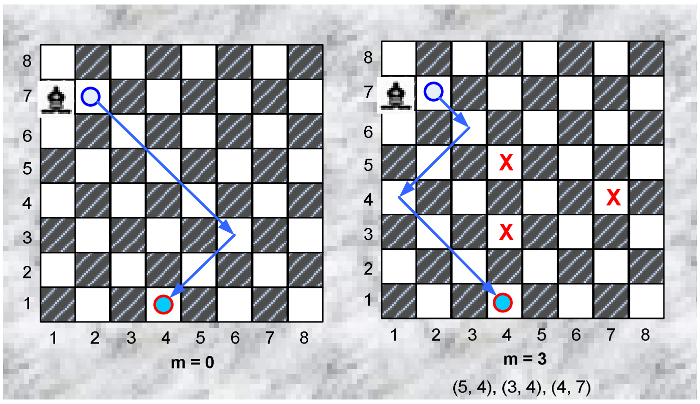
SELECT.INP	SELECT.OUT
3 -1 -4 7 9 9 5 8 7 3 -6 9 2	32

SELECT.INP	SELECT.OUT
3 5 5 5 5 5 5 5 5 5 5 5 5	30

Hạn chế: Trong tất cả các test: $n \le 10000$, $|a_{ij}| \le 30000$. Có 50% số lượng test với $n \le 1000$.

Bài 2. Quân tượng

Xét bàn cờ vuông kích thước $n \times n$. Các dòng được đánh số từ 1 đến n, từ dưới lên trên. Các cột được đánh số từ 1 đến n từ trái qua phải. Ô nằm trên giao của dòng i và cột j được gọi là ô (i,j). Trên bàn cờ có m $(0 \le m \le n)$ quân cờ. Với $m > 0$, quân cờ thứ i ở ô (r_i, c_i), $i = 1,2,..., m$. Không có hai quân cờ nào ở trên cùng một ô. Trong số các ô còn lại của bàn cờ, tại ô (p, q) có một quân tượng. Mỗi một nước đi, từ vị trí đang đứng quân tượng chỉ có thể di chuyển đến được những ô trên cùng đường chéo với nó mà trên đường đi không phải qua các ô đã có quân.



Cần phải đưa quân tượng từ ô xuất phát (p, q) về ô đích (s,t). Giả thiết là ở ô đích không có quân cờ. Nếu ngoài quân tượng không có quân nào khác trên bàn cờ thì chỉ có 2 trường hợp: hoặc là không thể tới được ô đích, hoặc là tới được sau không quá 2 nước đi (hình trái). Khi trên bàn cờ còn có các quân cờ khác, vấn đề sẽ không còn đơn giản như vậy.

Yêu cầu: Cho kích thước bàn cờ n, số quân cờ hiện có trên bàn cờ m và vị trí của chúng, ô xuất phát và ô đích của quân tượng. Hãy xác định số nước đi ít nhất cần thực hiện để đưa quân tượng về ô đích hoặc đưa ra số -1 nếu điều này không thể thực hiện được.

Dữ liệu: Vào từ file văn bản BISHOP.INP:

- Dòng đầu tiên chứa 6 số nguyên n, m, p, q, s, t;
- Nếu $m > 0$ thì mỗi dòng thứ i trong m dòng tiếp theo chứa một cặp số nguyên r_i , c_i xác định vị trí quân thứ i.

Hai số liên tiếp trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Kết quả: Đưa ra file văn bản BISHOP.OUT một số nguyên là số nước đi tìm được.

Ví dụ:

BISHOP.INP	BISHOP.OUT
8 3 7 2 1 4 5 4 3 4 4 7	3

Hạn chế: Trong tất cả các test: $1 \le n \le 200$. Có 60% số lượng test với $n \le 20$.

Bài 3. Kênh xung yếu

Một hệ thống n máy tính (các máy tính được đánh số từ 1 đến n) được nối lại thành một mạng bởi m kênh nối, mỗi kênh nối hai máy nào đó và cho phép truyền tin một chiều từ máy này đến máy kia. Ta gọi một ***mạch vòng*** của mạng đã cho là một dãy các máy tính và các kênh nối chúng có dạng:

$$u_1, e_1, u_2, \ldots, u_i, e_i, u_{i+1}, \ldots, u_{k-1}, e_{k-1}, u_k, e_k, u_1$$

trong đó $u_1, u_2, \ldots, u_k$ là các máy tính khác nhau trong mạng, e_i – kênh truyền tin từ máy u_i đến máy u_{i+1} ($i = 1, 2, \ldots, k$-1), e_k là kênh truyền tin từ máy u_k đến máy u_1. Một kênh truyền tin trong mạng được gọi là ***kênh xung yếu*** nếu như bất cứ mạch vòng nào của mạng cũng đều chứa nó.

Yêu cầu: Hãy xác định tất cả các kênh xung yếu của mạng đã cho.

Dữ liệu: Vào từ file văn bản CIRARC.INP:

- Dòng đầu tiên chứa 2 số nguyên dương n và m.
- Dòng thứ i trong số m dòng tiếp theo mô tả kênh nối thứ i bao gồm hai số nguyên dương u_i, v_i cho biết kênh nối thứ i cho phép truyền tin từ máy u_i đến máy v_i.

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

Kết quả: Ghi ra file văn bản CIRARC.OUT:

- Dòng đầu tiên ghi số nguyên k là số lượng kênh xung yếu trong mạng đã cho. Ghi k = -1 nếu mạng không chứa kênh xung yếu.
- Nếu k>0 thì mỗi dòng trong số k dòng tiếp theo ghi thông tin về một kênh xung yếu tìm được theo qui cách mô tả giống như trong file dữ liệu vào.

Ví dụ:

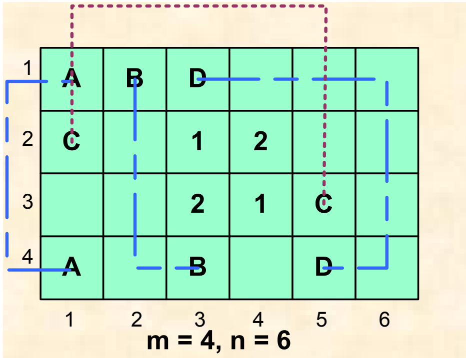
CIRARC.INP	CIRARC.OUT
2 2 1 2 2 1	2 1 2 2 1

CIRARC.INP	CIRARC.OUT
3 3 1 2 2 3 1 3	-1

Hạn chế: Trong tất cả các test: $n \leq 1000$, $m \leq 20000$. Có 50% số lượng test với $n \leq 200$.

Bài 4. Biến đổi bảng

Cho một bảng hình chữ nhật kích thước $m \times n$ ô vuông kích thước đơn vị. Các dòng được đánh số từ 1 đến m, từ trên xuống dưới. Các cột được đánh số từ 1 đến n, từ trái qua phải. Mỗi ô của bảng hoặc được để trống hoặc chứa một ký tự lấy từ tập Σ gồm các chữ số từ 0 đến 9 và các chữ cái la tinh in hoa từ A đến Z. Hai ô chứa cùng một ký tự được gọi là giống nhau. Mỗi ký tự của tập Σ xuất hiện ở không quá 4 ô trong bảng. Hai ô giống nhau có thể xoá được nếu chúng có cạnh chung hoặc có thể nối các tâm (giao điểm của hai đường chéo) của chúng với nhau bằng một đường gấp khúc gồm không quá 3 đoạn thẳng độ dài nguyên, mỗi đoạn song song với cạnh của bảng và ngoại trừ hai ô cần xoá, đường gấp khúc chỉ qua các ô trống hay nằm ngoài bảng. Các ô bị xoá trở thành ô trống. Mỗi lần xoá một cặp ô của bảng được gọi là một bước. Hình bên nêu ví dụ với trường hợp m = 4 và n = 6. Bước đầu tiên có thể xoá hai ô chứa ký tự '*A*', tiếp theo, lần lượt xoá các cặp ô chứa '*B*', chứa '*C*' và cặp ô chứa '*D*'. Ở ví dụ này, sau khi thực hiện 4 bước xoá có thể, trong bảng còn lại 4 ô không thể xoá được.



Yêu cầu: Cho m, n và m xâu độ dài n mô tả các dòng của bảng. Hãy xác định số lượng ô lớn nhất có thể xoá được.

Dữ liệu: Vào từ file văn bản CHANGE.INP:

- Dòng đầu tiên chứa 2 số nguyên m, n được ghi cách nhau bởi dấu cách.
- Dòng thứ i+1 chứa xâu n ký tự mô tả dòng thứ i của bảng (i = 1, 2, ..., m). Các ô trống được thể hiện bằng dấu chấm ('.').

Kết quả: Đưa ra file văn bản CHANGE.OUT một số nguyên là số lượng ô lớn nhất có thể xoá được.

Ví dụ:

CHANGE.INP	CHANGE.OUT
4 5 ABD... C.12.. ..21C. A.B.D.	8

CHANGE.INP	CHANGE.OUT
4 6 ABCDUV BADCVU ABCDUV BADCVU	24

Hạn chế: Trong tất cả các test: $0 < m \le 10$, $0 < n \le 10$. Có 60% số lượng test có $m \le 5$, $n \le 6$ và số lượng các ô khác trống không quá $\frac{m \times n}{2}$.

Năm 2006

Bài 1. Phân cụm

Phân cụm là một bài toán có ý nghĩa ứng dụng quan trọng trong các lĩnh vực như khai phá dữ liệu, thu thập dữ liệu và đòi hỏi phân hoạch tập các điểm dữ liệu ra thành các nhóm sao cho các điểm trong cùng một nhóm là "gần nhau" và "cách xa" các nhóm khác. Trong bài này chúng ta xét một dạng đơn giản của bài toán phân cụm. Cho tập gồm n đối tượng $X = \{x_1, x_2, ..., x_n\}$, khoảng cách $d(x_i, x_j)$ giữa mọi cặp $x_i \neq x_j$ và một số nguyên dương k ($k \leq n$). Giả thiết rằng: $d(x_i, x_j)$ là các số nguyên dương, $d(x_i, x_j) = d(x_j, x_i)$ và $d(x_i, x_i) = 0$, với mọi $i, j = 1, 2, ..., n$. Ta gọi một cách phân cụm là một cách phân hoạch tập X ra thành k tập con khác rỗng (mỗi tập con như vậy được gọi là một cụm). Cho $\mathbb{C} = \{C_1, C_2, ..., C_k\}$ là một cách phân cụm, ta gọi ***độ phân tách*** của cách phân cụm $\mathbb{C}$ (ký hiệu là $\rho(\mathbb{C})$) là giá trị nhỏ nhất trong số các khoảng cách giữa hai phần tử bất kỳ thuộc hai cụm khác nhau, nghĩa là

$$\rho(\mathbb{C}) = \min \{d(u,v): u \in C_p, v \in C_q, p \neq q\}.$$

Yêu cầu: Tìm cách phân cụm với độ phân tách là lớn nhất.

Dữ liệu: Vào từ file văn bản CLUSTER.INP:

- Dòng đầu tiên chứa hai số nguyên n và k.
- Dòng thứ i trong số n dòng tiếp theo ghi các số $d(x_i, x_1), d(x_i, x_2), ..., d(x_i, x_n)$, $i = 1, 2, ..., n$.

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

Kết quả: Ghi ra file văn bản CLUSTER.OUT độ phân tách của cách phân cụm tìm được.

Ví dụ:

CLUSTER.INP	CLUSTER.OUT
4 3 0 1 2 3 1 0 2 3 2 2 0 3 3 3 3 0	2

Hạn chế:

- Trong tất cả các test: $1 < k \leq n \leq 200$; $d(x_i, x_j) \leq 32000$, $i, j = 1, 2, ..., n$.
- Có 50% số lượng test với $n \leq 100$.

Bài 2. Các cửa hàng

Chủ chung cư Sao Khuê cho thuê dãy N cửa hàng được đánh số từ 1 đến N từ trái qua phải. Năm đầu tiên giá thuê cửa hàng thứ i là a_i đồng. Theo dòng thời gian, có cửa hàng làm ăn phát đạt, có cửa hàng làm ăn thua lỗ. Chủ các cửa hàng làm ăn phát đạt tìm cách thuê thêm diện tích của cửa hàng liền kề để mở rộng công việc kinh doanh. Như vậy có cửa hàng phải đóng cửa, có cửa hàng được mở rộng. Khi một cửa hàng i phát triển và thuê thêm diện tích của cửa hàng j liền kề nó ($j=i-1$ hoặc $j=i+1$), tiền thuê cửa hàng i sẽ được tính theo qui tắc sau:

- Nếu a_i mod 3 = 0, thì tiền thuê mới là $a_i + a_j$;
- Nếu a_i mod 3 = 1, thì tiền thuê mới là $2a_i + (a_j$ div 2);
- Nếu a_i mod 3 = 2, thì tiền thuê mới là $4a_i + (a_j$ div 4).

Theo luật của chủ chung cư Sao Khuê, mỗi một năm chỉ có không quá một cửa hàng được phép mở rộng và khi có cửa hàng mở rộng dãy các cửa hàng được đánh số lại bắt đầu từ 1 từ trái qua phải. Sau $N-1$ năm, chủ chung cư Sao Khuê giật mình khi phát hiện ra rằng chỉ còn lại duy nhất một cửa hàng ở chung cư. Để tìm hiểu quá trình phát triển của cửa hàng, chủ chung cư đã tìm đến sổ sách. Đáng tiếc là do sự bất cẩn của kế toán, sổ sách ghi nhận chi tiết đã bị thất lạc, chỉ còn lại giá thuê các cửa hàng trong năm đầu tiên và năm thứ $N-1$.

Yêu cầu: Hãy giúp chủ chung cư Sao Khuê khôi phục thông tin về quá trình phát triển của cửa hàng hiện nay.

Dữ liệu: Vào từ file văn bản SHOPS.INP:

- Dòng đầu tiên ghi số N;
- Dòng thứ hai chứa dãy số nguyên dương $a_1, a_2, ..., a_N$;
- Dòng thứ ba chứa số nguyên là số tiền thuê cửa hàng hiện nay.

Kết quả: Ghi ra file văn bản SHOPS.OUT gồm $N-1$ dòng, dòng thứ i chứa hai số nguyên theo thứ tự là chỉ số của cửa hàng mở rộng và cửa hàng đóng cửa trong năm thứ i, i=1, 2, ..., $N-1$. (Nếu có nhiều cách khôi phục, chỉ cần đưa ra một cách tùy ý.)

Ví dụ:

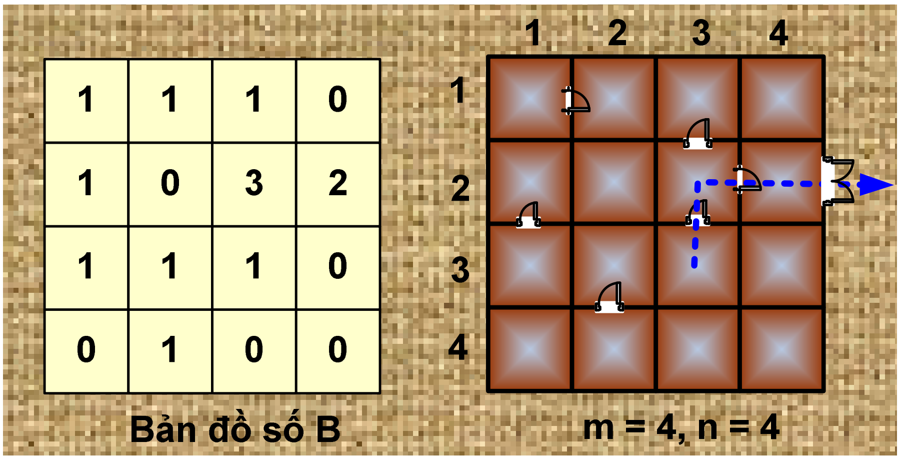
SHOPS.INP	SHOPS.OUT
3 3 2 4 21	1 2 1 2

Hạn chế:

- Giả thiết là dữ liệu đảm bảo luôn có ít nhất một cách khôi phục.
- Trong tất cả các test: $1 \le N \le 30$; $a_i \le 32000$, i =1, 2, ..., N.
- Có 50% số lượng test với $N \le 15$.

Bài 3. Thoát mê cung

Mê cung có dạng lưới ô vuông hình chữ nhật kích thước $m \times n$, các dòng của lưới được đánh số từ 1 đến m từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái sang phải. Ô nằm trên giao của dòng i và cột j được gọi là ô (i, j). Mỗi ô là một phòng. Vách ngăn giữa hai phòng hoặc giữa phòng với phần bên ngoài có thể có hoặc không có cửa thông nhau. Mê cung được mô tả bởi bản đồ số B là bảng $m \times n$ số nguyên, trong đó thành phần ở vị trí giao của dòng i với cột j là B_{ij} ($0 \le B_{ij} \le 4$) cho biết số vách ngăn có cửa của ô (i, j). Thời gian đi từ một ô sang ô bên cạnh hoặc ra ngoài là 1, nếu vách ngăn tương ứng có cửa.



Yêu cầu: Cho biết m, n, bản đồ số B và u, v – toạ độ một ô nào đó trong mê cung. Hãy xác định một cách đặt cửa cho các phòng đảm bảo thỏa mãn bản đồ B và thời gian T để thoát ra khỏi mê cung từ phòng (u, v) là ít nhất. Biết rằng dữ liệu bản đồ số B đảm bảo có ít nhất một cách đặt cửa để thoát được ra ngoài.

Dữ liệu: Vào từ file văn bản ESCAPE.INP:

- Dòng đầu tiên chứa 4 số nguyên m, n, u và v;
- Dòng thứ i trong m dòng tiếp theo chứa n số nguyên $B_{i1}, B_{i2}, \ldots, B_{in}$.

Các số trên cùng một dòng được phân tách nhau bởi dấu cách.

Kết quả: Đưa ra file văn bản ESCAPE.OUT số nguyên T.

Ví dụ:

ESCAPE.INP	ESCAPE.OUT
4 4 3 3 1 1 1 0 1 0 3 2 1 1 1 0 0 1 0 0	3

Hạn chế:

- Trong tất cả các test: $2 \le m, n \le 50$.
- Có 50% số lượng test với $m, n \le 25$.

Bài 4. Bumêran

Bumêran là công cụ đi săn bằng đá hoặc gỗ của thổ dân Úc. Nếu ném một cách thích hợp, bumêran sẽ bay tới đích rồi lượn về tay người ném. Để đáp ứng nhu cầu thể thao rèn luyện kỹ năng sử dụng bumêran, một công ty quyết định chế tạo bumêran bằng nhựa. Mỗi khi được ném với một lực nào đó về phía trước, bumêran sẽ bay theo một quĩ đạo nhất định và không nhất thiết quay trở về vị trí ban đầu. Theo thiết kế, mỗi bumêran được gắn với một xâu *S* gồm không quá 250 ký tự lấy từ tập {'F','R'}. Quĩ đạo bay của mỗi cách ném bumêran dọc theo đường thẳng đều có thể mô tả được bởi xâu ký tự nhận được bằng cách xoá bớt một số ký tự của xâu *S* gắn với nó, trong đó mỗi ký tự cho biết bumêran bay qua một khoảng độ dài 1 về hướng nào:

- F – bumêran bay về phía trước,
- R – bumêran bay ngược trở lại.

Một cách ném bumêran dọc theo đường thẳng được gọi là an toàn nếu thoả mãn 2 điều kiện:

- Điều kiện Bumêran: vị trí bắt đầu và kết thúc của quĩ đạo bay là trùng nhau.
- Điều kiện an toàn: Nếu vị trí ban đầu xuất hiện trên quĩ đạo bay và quá trình bay còn chưa kết thúc thì hướng bay tiếp theo phải là 'F'.

Hai cách ném gọi là khác nhau nếu hai xâu mô tả quĩ đạo bay của chúng là khác nhau. Với mỗi bumêran, có thể có nhiều cách ném an toàn khác nhau. Ví dụ, với bumêran có xâu *S* = 'FFFRRRRRF', ta có 3 cách ném an toàn: 'FR', 'FFRR' và 'FFFRRR'.

Lưu ý là trong quĩ đạo bay của bumêran vị trí ban đầu có thể xuất hiện nhiều lần.

Yêu cầu: Cho xâu *S*, hãy xác định số cách ném an toàn khác nhau có thể thực hiện.

Dữ liệu: Vào từ file văn bản BUMERAN.INP chứa xâu *S*.

Kết quả: Đưa ra file văn bản BUMERAN.OUT một số nguyên là số lượng cách ném an toàn có thể thực hiện.

Ví dụ:

BUMERAN.INP	BUMERAN.OUT
FFFRRRRRF	3

Hạn chế: Có 50% số lượng test với xâu *S* có độ dài không quá 100.

Bài 5. Nhóm hình đồng dạng

Trên mặt phẳng cho K hình, đánh số thứ tự từ 1 đến K. Mỗi hình tạo bởi một đường gấp khúc khép kín không tự cắt có các cạnh song song với các trục tọa độ.

Ta nói hai hình là thuộc cùng một nhóm đồng dạng nếu bằng cách thực hiện các phép biến đổi: lấy đối xứng qua trục, xoay, tịnh tiến song song với trục, phóng to, thu nhỏ đối với một hình có thể đặt nó trùng khít lên hình kia.

Yêu cầu: Xác định số lượng nhóm đồng dạng và số lượng hình trong mỗi nhóm tìm được.

Dữ liệu: Vào từ file văn bản GROUP.INP:

- Dòng đầu tiên ghi số nguyên K là số lượng hình;
- Dòng thứ i trong K dòng tiếp theo mô tả hình thứ i: đầu tiên là k_i ($4 \le k_i \le 200$) là số lượng đỉnh của hình thứ i ($1 \le i \le K$), tiếp theo là k_i cặp tọa độ của các đỉnh được liệt kê theo một chiều đi vòng quanh hình (các tọa độ là các số nguyên có trị tuyệt đối không quá 20000).

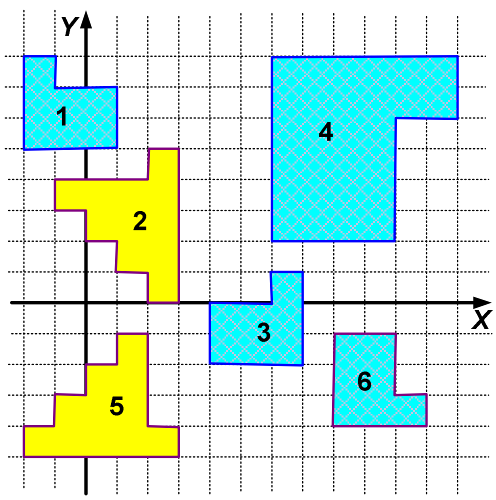
Các số trên cùng một dòng được ghi cách nhau một dấu cách.

Kết quả: Ghi ra file văn bản GROUP.OUT:

- Dòng đầu tiên ghi số nguyên M là số lượng nhóm hình đồng dạng tìm được.
- Dòng thứ hai ghi dãy M số nguyên là dãy số lượng các phần tử của các nhóm được liệt kê theo thứ tự không giảm.

Ví dụ:

GROUP.INP	GROUP.OUT
6 6 -2 8 -1 8 -1 7 1 7 1 5 -2 5 12 2 5 3 5 3 0 2 0 2 1 1 1 1 2 0 2 0 3 -1 3 -1 4 2 4 6 7 -2 4 -2 4 0 6 0 6 1 7 1 6 10 6 12 6 12 8 6 8 6 2 10 2 12 1 -1 1 -2 0 -2 0 -3 -1 -3 -1 -4 -2 -4 -2 -5 3 -5 3 -4 2 -4 2 -1 6 8 -1 8 -4 11 -4 11 -3 10 -3 10 -1	2 2 4



Hạn chế:

- Trong tất cả các test: $1 \le K \le 200$.
- Có 50% số lượng test với $K \le 100$.

Bài 6. Công viên Disneyland

Bờm và Cuội được mời đến vui chơi miễn phí tại công viên Disneyland nhân dịp công viên này được khai trương tại thành phố Mặt trăng. Trong công viên có tất cả n tụ điểm vui chơi (được đánh số từ 1 đến n). Trước khi đi Bờm và Cuội đã hứa sẽ chụp ảnh tất cả các tụ điểm có trong công viên để giới thiệu cho các bạn cùng lớp. Vì số lượng tụ điểm quá lớn mà thời gian lại hạn hẹp, Bờm và Cuội quyết định thực hiện việc dạo qua tất cả các tụ điểm theo qui tắc sau đây:

- Cùng xuất phát tại tụ điểm 1.
- Mỗi tụ điểm phải được ít nhất một người dạo qua.
- Mỗi người sẽ thăm các tụ điểm theo thứ tự tăng dần của chỉ số.
- Khi kết thúc, mỗi người từ tụ điểm cuối cùng trên đường đi của mình phải quay về nơi xuất phát mà không được đi qua bất cứ tụ điểm nào khác.

Yêu cầu: Cho biết t_{ij} là thời gian đi từ tụ điểm i đến tụ điểm j ($i, j = 1, 2, ..., n$), hãy xác định giúp Bờm và Cuội cách thực hiện cách đi tuân thủ qui tắc đã nêu sao cho tổng thời gian mà hai người phải đi là nhỏ nhất.

Dữ liệu: Vào từ file văn bản DISNEY.INP:

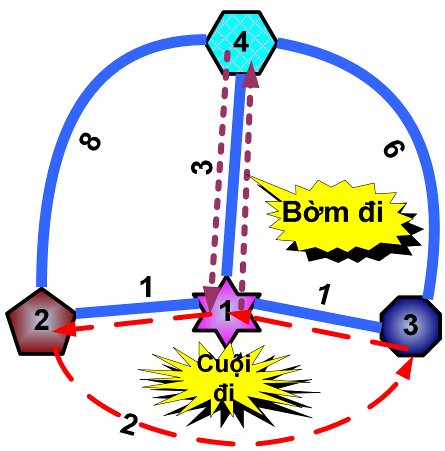
- Dòng đầu tiên chứa số nguyên dương n.
- Dòng thứ i trong số n dòng tiếp theo ghi các số nguyên $t_{i1}, t_{i2}, ..., t_{in}$, $i = 1, 2, ..., n$.

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

Kết quả: Ghi ra file văn bản DISNEY.OUT giá trị của tổng thời gian nhỏ nhất tìm được.

Ví dụ:

DISNEY.INP	DISNEY.OUT
4 0 1 1 3 1 0 2 8 1 2 0 6 3 8 6 0	10



Hạn chế:

- Trong tất cả các test: $1 \le n \le 200$; $0 \le t_{ij} \le 32000$, $i, j = 1, 2, ..., n$.
- Có 50% số lượng test với $n \le 100$.