

## Mục lục

Trò chơi thật giả — KIDGAME . . . . .	1
Mã sẵn tốt — GCHES . . . . .	3
Bày trận biển Đông — ESEA . . . . .	4
Chọn ngựa — HORSE . . . . .	6
Giá trị đường thay thế — RROAD . . . . .	8
Truy vấn trên bảng — TABQUERY . . . . .	10
Tìm số vắng mặt — missing . . . . .	11
Tô màu lưới — COLORTAB . . . . .	13
Chuỗi từ ghép — DECODE . . . . .	15

Nộp bài tại: [bkict.org:50002](http://bkict.org:50002)

Username: họ tên của mình viết rút gọn. Ví dụ: Nguyễn Hoàng Hải Minh  $\rightarrow$  *minhnhh*

Password như username.

## Bài A. Trò chơi thật giả

File dữ liệu vào:	Gọi hàm
File kết quả:	Trả về
Hạn chế thời gian:	0.1 giây
Hạn chế bộ nhớ:	512 MB

Thành thấy một đám trẻ con đang chơi một trò chơi rất thú vị. Thành lại gần bắt chuyện:

- Chào các cháu, các cháu đang chơi trò gì thế?
- Chúng cháu có một trọng tài, còn những người còn lại được chia vào hai đội: đội Sao và đội Trăng. Có  $n$  cháu trong đội Sao và  $m$  cháu trong đội Trăng. Chú hãy đoán xem ai trong số chúng cháu là trọng tài rồi chúng cháu sẽ nói cho chú biết chi tiết trò chơi?"
- Đồng ý, thế luật chơi thế nào?
- Chú có thể hỏi ai đó nằm trong đội nào nhưng không được phép hỏi trực tiếp ai đó có phải là trọng tài hay không. Biết rằng người trong đội Sao sẽ luôn nói thật, còn người trong đội Trăng luôn nói dối, còn trọng tài thì trả lời tùy theo thứ tự câu hỏi: luôn nói dối ở những câu hỏi lẻ: nghĩa là những câu hỏi thứ nhất, thứ 3, thứ 5, ..., và luôn nói thật ở những câu hỏi chẵn: nghĩa là những câu hỏi thứ 2, thứ 4, thứ 6, ....
- Thế có hạn chế gì không?
- Chú không được hỏi một người về chính bản thân người đó. Ví dụ chú không được hỏi câu: "Cháu ở đội Sao phải không?". Thêm nữa chú chỉ được phép hỏi một cháu tối đa là hai câu hỏi và hai câu hỏi này phải hỏi về hai cháu khác nhau. Ví dụ chú hỏi Dũng liệu Tuấn có thuộc đội Sao không, thì sau đó chú không được hỏi Dũng là liệu Tuấn có thuộc đội Trăng không.
- Nhất trí. Vậy chú sẽ tìm ra tất cả các cháu, mỗi cháu thuộc đội nào, không chỉ tìm ra trọng tài thôi đâu.

Chương trình phải sử dụng một thư viện riêng `kidgamelib.h` (cho C++). Trong chương trình của bạn cần khai báo các thư viện này ở đầu chương trình:

```
#include "kidgamelib.h"
```

Thư viện cung cấp các hàm sau:

- Các hàm khởi tạo trò chơi

- `int get_T();`

Chương trình phải gọi hàm này để khởi tạo trò chơi và chỉ được gọi duy nhất 1 lần. Hàm này trả về một giá trị  $T$  là số lượng bộ test.

- `void get_nm(&n,&m);`

Chương trình phải gọi hàm này  $T$  lần sau hàm `get_T()`, lần thứ  $i$  hàm này trả về hai số  $n$  và  $m$  tương ứng là số cháu trong đội Sao và Trăng tương ứng với bộ test thứ  $i$ . Các cháu được đánh số từ 1 đến  $n + m + 1$ .

- Hàm thực hiện truy vấn:

```
int ask(int a, int b, int d);
```

Hỏi cháu  $a$  liệu cháu  $b$  có nằm trong đội  $d$  không? Với  $1 \leq a, b \leq n + m + 1$ , và  $a \neq b$ .  $d = 0$  chỉ đội Sao và  $d = 1$  chỉ đội Trăng. Hàm trả về 1 nếu câu trả lời là 'Đúng', còn trả về 0 nếu câu trả lời là 'Sai'. Chỉ được phép hỏi một cháu về cùng một cháu khác tối đa 1 lần.

- Hàm trả lời câu hỏi:

```
void answer(int *C);
```

Các giá trị từ phần tử  $C[1]$  đến phần tử thứ  $C[n+m+1]$  chỉ vai trò từng cháu trong trò chơi: 0 nghĩa là thuộc đội Sao, 1 nghĩa là thuộc đội Trăng và 2 nghĩa là trọng tài. Phần tử  $C[0]$  không quan trọng, nhận giá trị tùy ý.

## Ví dụ

Gọi hàm	Trả về
<code>get_T()</code>	1
<code>get_nm(n,m)</code>	1 1
<code>ask(1,2,0)</code>	0
<code>ask(2,1,0)</code>	1
<code>ask(3,1,1)</code>	0
<code>answer(C)</code> với $C=[-1,2,1,0]$	Chương trình kết thúc

## Hạn chế

$$2 \leq n + m \leq 500$$

## Bài B. Mã săn tốt

©thai9cdb

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: 0.3 giây  
Hạn chế bộ nhớ: 512 Mb

Trên bàn cờ  $m \times n$  có  $k$  quân mã màu đen và một quân tốt màu trắng. Người chơi A và B lần lượt cầm quân đen và trắng, luân phiên thực hiện nước đi. Đến lượt mình, người chơi chọn một quân cờ của mình và thực hiện một nước đi hợp lệ. Cách di chuyển của các quân mã giống như luật cờ vua thông thường, còn quân tốt thì đi thẳng theo một cột của bàn cờ, và sẽ đổi hướng ngược lại nếu đi đến cạnh của bàn cờ. Tức là ban đầu nó đi theo chiều tăng của tọa độ dòng, ví dụ ban đầu tốt ở ô  $(i, j)$  thì nước đầu tiên nó đi là sang ô  $(i + 1, j)$ , tiếp theo là  $(i + 2, j)$ ,... cho đến khi chạm phải cạnh bàn cờ ( $i = m$ ) thì đổi sang hướng giảm tọa độ dòng, và cứ như vậy.

Bàn cờ cho phép có nhiều quân cờ đứng trong cùng một ô tại một thời điểm. Một quân mã được coi là bắt được quân tốt nếu nó di chuyển vào ô quân tốt đang đứng. Các quân mã được lập chiến thuật để bắt được quân tốt sớm nhất có thể.

Cho biết tọa độ của các quân cờ, hãy cho biết trước khi bị bắt, quân tốt đã di chuyển được bao nhiêu nước, hoặc thông báo là không thể bắt được dù chiến thuật chơi của người A tốt đến thế nào. Biết rằng quân trắng đi trước, lưu ý là nếu tốt di chuyển vào ô mã đang đứng thì sẽ không có chuyện gì xảy ra.

### Dữ liệu vào

- Dòng đầu tiên:  $m, n, k$
- Dòng tiếp theo ghi tọa độ ban đầu của tốt
- $k$  dòng tiếp theo ghi tọa độ của  $k$  quân mã

### Kết quả

Ghi ra một số nguyên là kết quả bài toán, ghi -1 nếu không thể bắt được.

### Ví dụ

stdin	stdout
8 8 1 1 1 4 2 10 10 1 10 1 8 1	1

### Hạn chế

- $4 \leq m, n \leq 1000$
- $1 \leq k < m * n - 1$
- Subtask 1 (30%):  $m, n \leq 50$
- Subtask 2 (30%):  $k = 1$
- Subtask 3 (40%): Ràng buộc gốc

## Bài C. Bày trận biển Đông

File dữ liệu vào:	<code>stdin</code>
File kết quả:	<code>stdout</code>
Hạn chế thời gian:	4 giây
Hạn chế bộ nhớ:	512 Mb

Vùng lãnh hải phía đông tổ quốc có  $n$  vùng biển trọng yếu. Toàn bộ vùng lãnh hải được mô tả trên bản đồ tọa độ, trong đó mỗi vùng biển trọng yếu được biểu diễn bởi một hình chữ nhật có các đỉnh góc là các điểm có tọa độ nguyên.

Nhằm chuẩn bị cho buổi tập trận “ESEA” lớn chưa từng có trên biển, ban chỉ huy quân sự Hải quân lập kế hoạch tác chiến trên bản đồ tọa độ nguyên mô phỏng vùng lãnh hải. Bộ đội Hải quân tiến hành đặt  $n$  cặp bộ dò, mỗi cặp bộ dò  $(\delta_1, \delta_2)$  tại hai điểm trọng yếu:

- bộ dò  $\delta_1$  đặt tại tọa độ  $(x_1, y_1)$  có khả năng phát hiện các vật thể trong phạm vi của góc phần tư bên dưới trái của nó, nghĩa là tất cả các điểm có tọa độ  $(u, v)$  thỏa mãn:  $u < x_1$  và  $v < y_1$ .
- bộ dò  $\delta_2$  tại tọa độ  $(x_2, y_2)$  có khả năng phát hiện các vật thể trong phạm vi của góc phần tư bên trên phải của nó, nghĩa là tất cả các điểm có tọa độ  $(u, v)$  thỏa mãn:  $u > x_2$  và  $v > y_2$ .

Hai cặp bộ dò  $i$  và  $j$  được gọi là có liên kết vẹn toàn với nhau nếu như cả hai bộ dò của cặp  $j$  nằm trọn vẹn trong phạm vi phát hiện của một trong hai bộ dò của cặp  $i$ .

**Yêu cầu:** Ban chỉ huy quân sự yêu cầu phân hoạch tập các cặp bộ dò ra thành ít nhất các nhóm sao cho mỗi cặp phải thuộc vào đúng một nhóm và trong mỗi nhóm không có hai cặp nào có liên kết vẹn toàn với nhau.

### Dữ liệu vào

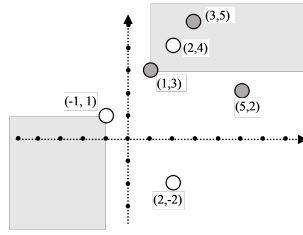
Dòng thứ nhất chứa duy nhất một số nguyên dương  $K$  ( $K \leq 20$ ) là số lượng bộ dữ liệu. Tiếp theo là  $K$  nhóm dòng, mỗi nhóm tương ứng với một bộ dữ liệu có cấu trúc như sau:

- Dòng đầu chứa số nguyên dương  $n$ ;
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo ghi 4 số nguyên  $x_1, y_1, x_2, y_2$  mỗi số nhỏ hơn  $10^9$  là tọa độ vị trí của hai bộ dò của cặp thứ  $i$  ( $i = 1, 2, \dots, n$ ).

### Kết quả

Ghi ra  $K$  nhóm dòng, mỗi nhóm dòng có khuôn dạng sau:

- Dòng đầu ghi số nguyên  $m$  là số lượng nhóm tìm được;
- Dòng thứ  $i$  trong số  $m$  dòng tiếp theo ghi một nhóm các cặp bộ dò đôi một không có liên kết vẹn toàn có dạng:
  - Đầu tiên ghi số nguyên  $t_i$  là số lượng cặp bộ dò trong nhóm;
  - $t_i$  số tiếp theo ghi chỉ số các cặp bộ dò trong nhóm.



Hình 1: Hình minh họa cho ví dụ thứ nhất.

## Ví dụ

stdin	stdout
3 -1 1 1 3 2 -2 5 2 2 4 3 5	2 2 1 2 1 3
2 -2 1 -2 1 -1 0 -1 0	1 2 1 2

## Giải thích

### Hạn chế

- Có 20% số lượng test thỏa mãn điều kiện:  $n \leq 10$ ;
- Có thêm 20% số lượng test thỏa mãn điều kiện:  $n \leq 1000$ ; và trong tất cả các cặp, hai bộ dò của chúng đặt tại cùng một điểm, nghĩa là  $x_1 = x_2, y_1 = y_2$ ;
- Có thêm 20% số lượng test thỏa mãn điều kiện:  $n \leq 10^6$ ; và trong tất cả các cặp, hai bộ dò của chúng đặt tại cùng một điểm, nghĩa là  $x_1 = x_2, y_1 = y_2$ ;
- Có thêm 20% số lượng test thỏa mãn điều kiện:  $n \leq 1000$ ;
- 20% số lượng test còn lại thỏa mãn điều kiện:  $n \leq 10^6$ ;

## Bài D. Chọn ngựa

File dữ liệu vào:	<code>stdin</code>
File kết quả:	<code>stdout</code>
Hạn chế thời gian:	0.1 giây
Hạn chế bộ nhớ:	512 MB

Ở thời Xuân Thu, hàng năm cứ mỗi khi mùa xuân đến Tề Vương lại tổ chức đua ngựa và yêu cầu quan tể tướng của mình là Điền Kỵ phải tham gia. Cuộc đua bao gồm 3 vòng đấu, ở vòng đấu thứ nhất hai bên thường đưa con ngựa tốt nhất của mình ra thi đấu, vòng 2 là các con ngựa tốt thứ nhì và vòng 3 là các con ngựa tốt thứ ba. Kết cục của mỗi vòng đấu chỉ có thắng thua, không có hòa. Ai thắng 2 trong số ba vòng đấu sẽ là người thắng chung cuộc. Phần thắng bao giờ cũng thuộc về Tề Vương do các con ngựa của Tề Vương ở các vòng đấu là tốt hơn hẳn các con ngựa của Điền Kỵ. Có một lần, nhờ sử dụng thuật của quân sư Tôn Tẫn bày cho, Điền Kỵ lần đầu tiên đã thắng Tề Vương trong cuộc đua ngựa. Sau khi thua ở cuộc đua, mặc dù rất tức giận nhưng vì đã hứa nếu thua sẽ thưởng cho Điền Kỵ một trong 3 con ngựa tốt nhất trong tàu ngựa của mình, nên Tề Vương đã nghĩ ra một cách để làm khó Điền Kỵ. Tề Vương đã truyền chỉ ban thưởng cho Điền Kỵ con ngựa tốt thứ  $k$  ( $1 \leq k \leq 3$ ) trong tàu ngựa của mình và cho phép Điền Kỵ đến gặp Mã Giám Quan để lấy ngựa. Điền Kỵ phấn khởi nhanh chóng đến gặp Mã Giám Quan để nhận ngựa, thế nhưng:

Trong tàu ngựa của Tề Vương có  $n$  con ngựa được đánh thứ tự từ 1 đến  $n$ , không có hai con nào có độ tốt như nhau. Tề Vương đã mật lệnh cho Mã Giám Quan không cho phép Điền Kỵ xem ngựa, không được phép nói gì mà chỉ được phép gật hoặc lắc tương ứng với câu trả lời khẳng định hoặc phủ định cho câu hỏi dạng: “Con ngựa thứ  $i$  tốt hơn con ngựa thứ  $j$  phải không?” (với  $1 \leq i, j \leq n$  và  $i \neq j$ ). Mã Giám Quan chỉ giao ngựa nếu như Điền Kỵ bằng việc thực hiện các câu hỏi dạng vừa nêu trên chọn được đúng con ngựa tốt thứ  $k$  và sau không quá một số tối đa lần hỏi cho phép (ký hiệu số này là  $d$ ). Trong trường hợp không giao ngựa, để Điền Kỵ tâm phục, khẩu phục, Mã Giám Quan sẵn sàng đưa ra dãy độ tốt của  $n$  con ngựa khớp với tất cả các câu trả lời cho các câu hỏi của Điền Kỵ, nhưng Điền Kỵ hoặc đã không xác định được đúng con ngựa tốt thứ  $k$  hoặc đã hỏi vượt quá  $d$  là số lượng câu hỏi tối đa được phép hỏi.

Đứng trước bài toán hóc búa này, một lần nữa Điền Kỵ phải nhờ đến Tôn Tẫn bày cách cho.

Yêu cầu: Hãy đóng vai Tôn Tẫn giúp Điền Kỵ đưa ra các câu hỏi để nhận được phần thưởng của Tề Vương.

Chương trình phải sử dụng một thư viện riêng. Thư viện bao gồm các file sau: `horselib.pas` (cho Pascal), `horselib.h` (cho C++). Trong chương trình của bạn cần khai báo các thư viện này ở đầu chương trình:

- `uses horselib;` đối với Pascal;
- `#include "horselib.h"` đối với C++.

Thư viện cung cấp các hàm sau:

- Các hàm khởi tạo trò chơi
  - `function get_n():longint;` đối với Pascal,  
`int get_n();` đối với C++.  
Chương trình phải gọi hàm này để khởi tạo trò chơi. Hàm này trả về một giá trị  $n$  là số lượng con ngựa trong tàu ngựa.
  - `function get_k():longint;` đối với Pascal,  
`int get_k();` đối với C++.  
Chương trình phải gọi hàm này ngay sau hàm `get_n()`. Hàm này trả về số  $k$  tương ứng với việc Điền Kỵ phải chọn con ngựa tốt thứ  $k$  trong tàu ngựa.
- Hàm thực hiện truy vấn:  
`function compare(i:longint; j:longint):longint;` đối với Pascal,

`int compare(int i, int j);` đối với C++,  
Hàm này trả về 1 nếu con ngựa thứ  $i$  tốt hơn con ngựa thứ  $j$  và 0 nếu ngược lại.

- Hàm trả lời câu hỏi

`procedure guess(res:longint);` đối với Pascal,

`void guess(int res);` đối với C++.

Để kết thúc chương trình cần gọi hàm này với  $res$  là số thứ tự của con ngựa tốt thứ  $k$  trong tàu ngựa. Sau khi gọi hàm này chương trình sẽ tự động kết thúc. Số lượng câu hỏi của chương trình của bạn sẽ bằng tổng số lần gọi hàm `compare(i,j)`.

Lưu ý: Mỗi hàm `get_n`, `get_k` và `guess` chỉ được gọi một lần duy nhất. Bạn có thể xem các file được cung cấp trên hệ thống để hiểu rõ hơn về cách tương tác với hệ thống.

## Ví dụ

stdin	stdout
<code>get_n()</code>	3
<code>get_k()</code>	2
<code>compare(1,2)</code>	0
<code>compare(1,3)</code>	1
<code>guess(1)</code>	Kết thúc chương trình. Chương trình đã trả lời đúng với số lần đặt câu hỏi là ít nhất.

## Hạn chế

- Subtask 1 : Có 20% số test ứng với  $n \leq 1000$ ;  $k = 1$ ; không cho biết  $d$ ;
- Subtask 2 : Có 40% số test ứng với  $n \leq 1000$ ;  $k = 2$ ; không cho biết  $d$ ;
- Subtask 3 : Có 40% số test ứng với  $900 \leq n \leq 1000$ ;  $k = 3$ ;  $d = 1017$ .

## Bài E. Giá trị đường thay thế

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: 1 giây  
Hạn chế bộ nhớ: 512 Mb

Thành phố Alpha có  $n$  đầu nút giao thông và  $m$  đoạn đường hai chiều, mỗi đoạn kết nối hai đầu nút. Các đầu nút được đánh số từ 1 đến  $n$ . Mỗi đoạn đường có độ dài là một số nguyên dương. Trong những giờ cao điểm các đoạn đường có mật độ giao thông tăng chóng mặt. Do đó, để đánh giá khả năng tìm đường thay thế tránh mỗi đoạn đường  $(x, y)$ , Ban giải pháp chống tắc nghẽn của thành phố mới đưa vào một đại lượng gọi là **giá trị thay thế**  $r_{x,y}$  được tính bởi công thức sau:

$$r_{xy} = \max_{u,v} \frac{f(u,v,x,y)}{g(u,v)}, \forall u, v = 1, \dots, n,$$

với:

- $f(u, v, x, y)$  là độ dài đường đi ngắn nhất từ  $u$  đến  $v$  khi mà đoạn  $(x, y)$  bị nghẽn không đi chuyển qua được.
- $g(u, v)$  là độ dài đường đi ngắn nhất từ  $u$  đến  $v$  mà không có đoạn đường nào bị nghẽn.

Đặt  $r_{xy} = -1$  nếu không có đường đi nào giữa hai đầu nút  $u, v$  bất kỳ khi đoạn  $(x, y)$  bị nghẽn. Nhắc lại độ dài đường đi từ  $u$  đến  $v$  là tổng tất cả độ dài các đoạn đường trên đường đi đó.

**Yêu cầu:** Hãy tìm đoạn đường có giá trị thay thế lớn nhất.

### Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên  $n, m$
- Dòng thứ  $i^{th}$  trong số  $m$  dòng tiếp theo chứa ba số nguyên dương  $x_i, y_i, d_i$ , ( $1 \leq x_i, y_i < n, d_i \leq 10^6$  với  $i = 1, 2, \dots, m$ ) mô tả một đoạn đường hai chiều có các đầu nút  $x_i$  và  $y_i$  với khoảng cách  $d_i$ .

### Kết quả

Ghi ra trên một dòng duy nhất giá trị  $r_{max}$  tìm được với sai số nhỏ hơn  $10^{-6}$ .

### Ví dụ

stdin	stdout
5 7 1 2 1 1 3 2 1 4 1 2 3 5 2 5 1 5 3 1 3 4 6	8.000000
3 2 1 2 8 2 3 5	1.000000
2 1 1 2 10	-1.000000



## Hạn chế

- Subtask 1: Có 20% số test ứng với  $(2 \leq n \leq 200, 1 \leq m \leq 2000)$ ;
- Subtask 2: Có 30% số test ứng với  $(2 \leq n \leq 1000, 1 \leq m \leq 2000)$ ;
- Subtask 3: Có 50% số test ứng với  $(2 \leq n \leq 1000, 1 \leq m \leq 10000)$ ;

## Bài F. Truy vấn trên bảng

©thai9cdb

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: 1 giây  
Hạn chế bộ nhớ: 512 Mb

Trên mặt phẳng tọa độ  $Oxy$  người ta tiến hành các thao tác nằm trong một trong hai dạng sau:

- 1  $x\ y\ m$ : Đặt 1 vật có khối lượng  $m$  vào điểm tọa độ  $(x, y)$
- 2  $x\ y\ x'\ y'$ : Tính tổng khối lượng các điểm  $(u, v)$  thỏa mãn:  $x \leq u \leq x', y \leq v \leq y'$

**Yêu cầu:** Với mỗi truy vấn loại 2, hãy đưa ra kết quả tổng khối lượng các điểm thỏa mãn truy vấn này.

### Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên dương  $Q \leq 40000$  là số lượng truy vấn;
- $Q$  dòng tiếp theo, mỗi dòng ghi một trong 2 định dạng truy vấn như trên với mỗi giá trị tọa độ được tính bởi hàm  $f(k) = (k + \delta) \% 10^9$ , với  $\delta$  là tổng giá trị tuyệt đối các kết quả của các truy vấn loại 2 ở trước truy vấn đang xét, ban đầu  $\delta = 0$ . Các giá trị được mô tả như sau:
  - 1  $x_0\ y_0\ m$ , là truy vấn loại 1 với các tham số tọa độ  $x$  và  $y$  được tính bởi  $f(x_0)$  và  $f(y_0)$ ;
  - 2  $x_0\ y_0\ x'_0\ y'_0$ , là truy vấn loại 2 với các tham số tọa độ  $x\ y\ x'\ y'$  tính bởi  $f(x_0)\ f(y_0)\ f(x'_0)\ f(y'_0)$ .

### Kết quả

Ghi ra lần lượt kết quả mỗi truy vấn loại 2 trên một dòng.

### Ví dụ

stdin	stdout
4 1 1 1 2 1 1 3 3 1 2 1 1 2 0 0 2 2	3

### Hạn chế

- Subtask 1: có 20% số test ứng với  $0 \leq x, y, x', y' \leq 1000$ .
- Subtask 2: có 30% số test ứng với  $0 \leq x, y, x', y' \leq 100000$ .
- Subtask 3: 50% số test còn lại ứng với  $0 \leq x, y, x', y' \leq 10^9$ .

Trong tất cả các Subtask  $m$  có giới hạn  $-50000 \leq m \leq 50000$ .

## Bài G. Tìm số vắng mặt

File dữ liệu vào:	Gọi hàm
File kết quả:	Trả về
Hạn chế thời gian:	0.1 giây
Hạn chế bộ nhớ:	256 MB

Alice và Bob rủ nhau chơi trò chơi tìm các số vắng mặt sau đây. Từ các phần tử của dãy số  $1, 2, \dots, n$  ( $2 \leq n \leq 1000$ ), Alice xây dựng tập  $X$  gồm  $m$  số  $a_1, a_2, \dots, a_m$  với  $a_1 < a_2 < \dots < a_m$ . Nhiệm vụ của Bob là tìm ra  $n - m$  số trong dãy  $1, 2, \dots, n$  không có mặt trong tập các số  $X$  mà Alice đang giữ, bằng cách yêu cầu Alice trả lời câu hỏi có dạng sau đây:

*“Bít thứ  $j$  trong biểu diễn nhị phân của số  $a_i$  là bao nhiêu?”*

Câu trả lời của Alice cho câu hỏi sẽ là 0 hoặc 1. Lưu ý là các bit của một số nguyên  $k$  được đánh số từ 0 đến  $\lfloor \log_2 k \rfloor$ , từ phải qua trái.

Yêu cầu: Hãy giúp Bob tìm ra  $n - m$  số không có mặt trong tập gồm  $m$  số mà Alice đang giữ với số lượng câu hỏi cần yêu cầu Alice trả lời càng ít càng tốt.

Chương trình của bạn phải sử dụng một thư viện riêng. Thư viện bao gồm các file sau: `missinglib.h` (cho C++). Trong chương trình của bạn các thư viện này cần được khai báo ở đầu chương trình:

```
#include "missinglib.h"
```

Thư viện cung cấp các hàm sau:

- Hàm khởi tạo trò chơi

```
int get_n():
```

```
int get_m():
```

Chương trình của bạn phải gọi các hàm này để khởi tạo trò chơi và lấy giá trị cho 2 biến  $n$  và  $m$ .

- Hàm thực hiện truy vấn

```
int ask(int i, int j);
```

Hàm này trả về 1 nếu bít thứ  $j$  của số  $a_i$  bằng 1 và trả về 0 nếu ngược lại.

- Hàm trả lời câu hỏi

```
void guess(vector<int> res);
```

Để kết thúc, chương trình của bạn cần gọi hàm này với `res` là kiểu vector trở đến  $n - m$  số tìm được đã được sắp xếp theo thứ tự tăng dần của giá trị. Sau khi gọi hàm này chương trình sẽ tự động kết thúc. Số lượng câu hỏi của chương trình của bạn sẽ bằng tổng số lần gọi hàm `ask( $i, j$ )`.

**Lưu ý:** Mỗi hàm `get_n`, `get_m` và `guess` chỉ được gọi một lần duy nhất.

Bạn có thể xem các file được cung cấp trên hệ thống để hiểu rõ hơn về cách tương tác với hệ thống.

### Ví dụ

Gọi hàm	Trả về
<code>get_n()</code>	2
<code>get_m()</code>	1
<code>ask(1,1)</code>	1
<code>guess(res) // với res=&lt;2&gt;</code>	Kết thúc chương trình. Bạn đã trả lời đúng với số lượng câu hỏi là 1 và chương trình đạt điểm của ví dụ này.

### Hạn chế

- Subtask 1: có 30% số test ứng với  $m = n - 1$ ;  $a_1 < a_2 < \dots < a_m$ .
- Subtask 2: có 30% số test ứng với  $m = n - 2$ ;
- Subtask 3: 40% số test còn lại ứng với  $0 \leq m \leq n$  và  $m \neq n - 1$ .

## Bài H. Tô màu lưới

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: 1 giây  
Hạn chế bộ nhớ: 256 MB

Cho lưới ô vuông kích thước  $m \times n$ . Các hàng 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 vuông thuộc hàng thứ  $i$  và cột thứ  $j$  có tọa độ  $(i, j)$ . Người ta tô các ô vuông bởi  $2 \times n$  màu có mã màu được đánh số từ 1 đến  $2 \times n$  sao cho mỗi màu đều được tô cho ít nhất một ô. Ký hiệu  $L_j$  là số lượng màu khác nhau được sử dụng để tô các ô trong cột thứ  $j$  ( $j = 1, 2, \dots, n$ ). Ta gọi *độ đa sắc* của lưới là giá trị  $\max_{1 \leq j \leq n} L_j$ .

Cho phép thực hiện việc hoán đổi màu của hai ô ở hai đỉnh đối diện trên đường chéo của hình chữ nhật kích thước  $2 \times 3$  bất kỳ. Mỗi phép hoán đổi được mô tả bởi bốn số nguyên  $(u, v, s, t)$  cho biết hai ô vuông  $(u, v)$  và  $(s, t)$  được hoán đổi màu.

**Yêu cầu:** Hãy xác định một dãy các phép hoán đổi màu để đưa lưới về trạng thái có độ đa sắc nhỏ nhất.

### Dữ liệu vào

Vào từ thiết bị vào chuẩn: Dòng thứ nhất chứa số nguyên dương  $T$  ( $T \leq 30$ ) là số lượng bộ dữ liệu. Mỗi nhóm dòng trong  $T$  nhóm dòng tiếp theo mô tả một bộ dữ liệu theo khuôn dạng sau:

- Dòng đầu tiên chứa 2 số nguyên  $m, n$  được ghi cách nhau bởi dấu cách;
- Dòng thứ  $i$  trong số  $m$  dòng tiếp theo chứa  $n$  số nguyên dương  $c_{i1}, c_{i2}, \dots, c_{in}$  được ghi cách nhau bởi dấu cách, trong đó  $c_{ij}$  là mã màu của ô  $(i, j)$  trong lưới ban đầu ( $j = 1, 2, \dots, n$ ).

### Kết quả

Ghi ra thiết bị ra chuẩn T nhóm dòng, mỗi nhóm là kết quả tìm được cho bộ dữ liệu tương ứng trong dữ liệu vào, theo khuôn dạng sau:

- Dòng đầu tiên ghi ra số nguyên không âm  $p$  là số lượng phép hoán đổi cần thực hiện;
- Tiếp đến là  $p$  dòng mô tả dãy các phép hoán đổi cần thực hiện để đưa lưới về trạng thái có độ đa sắc nhỏ nhất. Mỗi dòng ghi 4 số nguyên dương  $u, v, s, t$  cách nhau bởi dấu cách cho biết cần thực hiện việc hoán đổi màu của hai ô vuông  $(u, v)$  và  $(s, t)$ .

Nếu có nhiều cách thực hiện để đưa lưới về trạng thái có độ đa sắc nhỏ nhất thì chỉ cần đưa ra một cách.

### Ví dụ

stdin	stdout
2	0
4 4	2
1 2 3 4	2 2 4 3
5 6 7 8	2 2 4 1
1 2 3 4	
5 6 7 8	
4 4	
1 2 3 4	
5 7 7 8	
1 2 3 4	
6 6 5 8	

## Hạn chế

- Subtask 1:  $4 \leq m, n \leq 5$ ;
- Subtask 2:  $6 \leq m, n \leq 50$ .

## Bài I. Chuỗi từ ghép

©thai9cdb

File dữ liệu vào: `stdin`  
File kết quả: `stdout`  
Hạn chế thời gian: 0.1 giây  
Hạn chế bộ nhớ: 512 Mb

Cho  $W$  là một tập các từ, mỗi từ tạo bởi một hoặc nhiều kí tự viết liền nhau. Từ tập  $W$ , ta có thể tạo ra các chuỗi bằng cách viết liền các từ của  $W$  (mỗi từ có thể sử dụng nhiều lần). Ví dụ chuỗi:  $S = x_1x_2 \dots x_k, x_i \in W, \forall i = \overline{1, k}$

Khi đó tập  $X = \{x_1, x_2, \dots, x_k\}$  được gọi là một dẫn xuất của chuỗi  $S$ . Rõ ràng là một chuỗi có thể có nhiều dẫn xuất, ví dụ:

$W = \{ab, ba, a\}$   
 $S = aba = (ab)a = a(ba)$

**Yêu cầu:** Hãy tìm một chuỗi có nhiều hơn một dẫn xuất. Hai dẫn xuất  $X = \{x_1, x_2, \dots, x_k\}$  và  $Y = \{y_1, y_2, \dots, y_k\}$  được coi là khác nhau nếu hoặc  $k \neq q$  hoặc  $\exists i \in \{1, \dots, k\} : x_i \neq y_i$ .

### Dữ liệu vào

- Dòng đầu tiên chứa một số nguyên dương  $n$  là lực lượng tập  $W$ ;
- Mỗi dòng trong số  $n$  dòng tiếp theo chứa một từ của  $W$ . Các từ chỉ chứa các chữ cái latin in thường, có độ dài nguyên dương và  $\leq 1000$ . Tổng độ dài các từ không quá 1000 và không có 2 từ nào giống nhau.

### Kết quả

Nếu không tồn tại chuỗi nào có nhiều hơn 1 dẫn xuất, in ra -1. Ngược lại, in ra 2 dòng mô tả 2 dẫn xuất, mỗi dòng là chuỗi các từ của một dẫn xuất phân cách nhau bởi kí tự '+'. Xem thêm trong test ví dụ để rõ thêm về định dạng output. Nếu có nhiều hơn 2 dẫn xuất thì chỉ cần in ra 2 dẫn xuất bất kỳ.

### Ví dụ

stdin	stdout
3	a+ba
ab	ab+a
ba	
a	

### Hạn chế

- Subtask 1: có 20% số test ứng với  $n \leq 10$ .
- Subtask 2: 80% số test còn lại ứng với giới hạn trong mô tả.