**KHOÁ HỌC: THUẬT TOÁN CHUYÊN SÂU – SAMSUNG PTIT – HÈ 2025**

***Thứ 7, ngày 17 tháng 08 năm 2025***

**Buổi 7: CONSTRUCTIVE ALGORITHM**

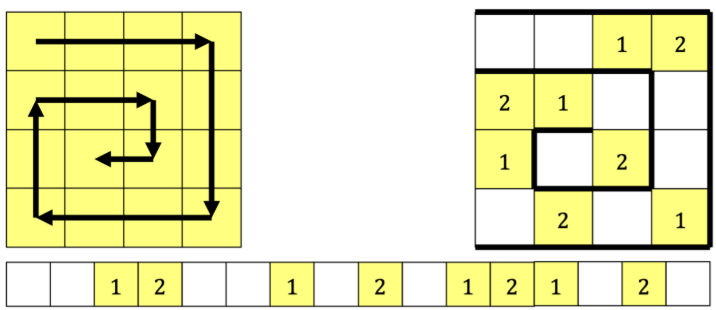
**ĐỀ BÀI**

**BÀI 1. BẢNG SỐ ĐẸP - 1**

Xét một bảng kích thước n dòng, n cột chứa các ô trống và các số tự nhiên 1 hoặc 2. Bảng số được gọi là đẹp nếu như nó có các tính chất:

* Mỗi hàng và mỗi cột chứa đúng một số 1, một số 2 và n – 2 ô trống
* Bắt đầu từ góc trên cùng bên trái, bắt đầu di chuyển sang phải rồi theo hình xoắn ốc vào trong, các giá trị ở các ô chứa số sẽ xuất hiện theo thứ tự 1, 2, 1, 2, …

Hình dưới đây là ví dụ về một bảng số đẹp kích thước n x n



Cho số n, vẽ bảng số đẹp kích thước n x n

**Input.** Chỉ có một số n (2 ≤ n ≤ 500)

**Ouput.**

* Nếu không có lời giải, chỉ ghi -1.
* Nếu có lời giải, in ra bất kỳ lời giải nào hợp lệ:
* Ghi ra n dòng của bảng số tìm được, mỗi dòng gồm n số, các số cách nhau đúng 1 dấu cách
* Những ô trống không có số thì ô đó nhận giá trị là 0

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | 0 0 1 2 2 1 0 0 1 0 2 0 0 2 0 1 |

**Chú ý.** Đây chỉ là một trong những đáp án hợp lệ. Học sinh hoàn toàn có thể có kết quả khác, miễn thoả mãn yêu cầu đề bài

**Subtask**

* **Subtask 1 (50%):** n ≤ 15
* **Subtask 2 (50%):** Không có điều kiện gì thêm

**Giới hạn thời gian:** 1.0s

**Giới hạn bộ nhớ:** 500M

**BÀI 2. OLIU VECTOR**

Oliu tạo ra 2n vector, mỗi vector gồm n thành phần, mỗi thành phần chỉ nhận giá trị 1 hoặc

−1, không có hai vector nào bằng nhau. Sau đó cậu chọn ngẫu nhiên một số thành phần của một số vector và thay bằng 0.

Oliu đã chọn được một tập các vector mà tổng các vector đó là zero vector (tất cả các thành phần đều bằng 0). Oliu thách đố bạn tìm được một tập các vector (có ít nhất một vector) mà tổng các vector là zero vector.

**Input**

* Dòng đầu tiên chứa số nguyên n (n ≤ 20);
* Tiếp theo là 2ndòng, mỗi dòng chứa n số nguyên mô tả một vector.

**Output.** Gồm một dòng là một xâu gồm 2𝑛 kí tự, kí tự thứ i bằng 1 tương ứng vector thứ i

được chọn, ngược lại kí tự thứ i bằng 0.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  0 -1  -1 1  1 0  1 1 | 1110 |

**Subtask:** Có tổng cộng 40 test, với mỗi n từ 1 đến 20 có 2 test.

**Giới hạn thời gian:** 3.0s

**Giới hạn bộ nhớ:** 500M

**BÀI 3. TÔ MÀU LƯỚI**

Cho lưới ô vuông kích thước m × 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 ivà cột thứ j có tọa độ (i, j). Người ta tô các ô vuông bởi nmàu có mã màu được đánh số từ 1 đến n. Ký hiệu Ljlà số lượng màu khác nhau được sử dụng để tô các ô trong cột thứ j (j= 1, 2, …, n). Ta gọi ***độ đa sắc*** của lưới là giá trị max {L1, L2, …, Ln}

*j*

Cho phép thực hiện việc hoán đổi màu của hai ô bất kỳ trên lưới. 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.

**X**ác định một dãy các phép hoán đổi màu để đưa lưới về trạng thái có độ đa sắc nhỏ nhất.

**Input**

Dòng thứ nhất chứa số nguyên dương T (T ≤ 30) là số lượng bộ test. Mỗi test gồm m + 1 dòng trong đó:

* Dòng đầu tiên chứa 2 số nguyên m, n (m, n ≤ 50) được ghi cách nhau bởi dấu cách;
* Dòng thứ i trong số mdòng tiếp theo chứa *n* số nguyên dương ci1, ci2, …, cinđược ghi cách nhau bởi dấu cách, trong đó cij là mã màu của ô (i, j) trong lưới ban đầu (j= 1, 2, …, n).

**Output**

Với mỗi test, in ra kết quả theo dạng sau:

* Dòng đầu tiên ghi ra số nguyên không âm plà 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**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2 3  1 2 3  3 1 2 | 2  2 1 2 2  2 2 2 3 |

**Giới hạn thời gian:** 1.0s

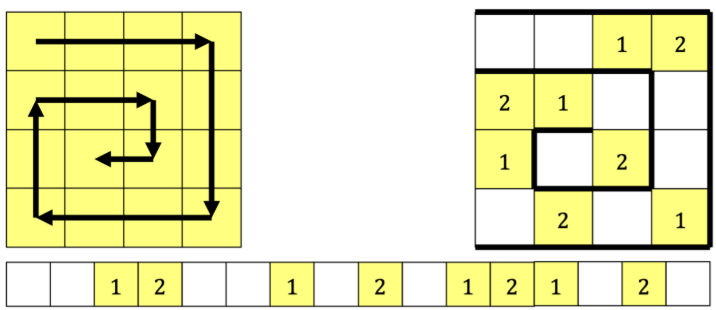
**Giới hạn bộ nhớ:** 500M

**BÀI 4. BẢNG SỐ ĐẸP – 2**

Xét một bảng kích thước n dòng, n cột chứa các ô trống và các số tự nhiên trong khoảng từ 1 đến k (k ≤ n). Bảng số được gọi là k - đẹp nếu như nó có các tính chất:

* Mỗi hàng và mỗi cột chứa đúng n – k số ô trống.
* Trên mỗi hàng và mỗi cột, mỗi số x (1 ≤ x ≤ k) xuất hiện đúng một lần.
* Bắt đầu từ góc trên cùng bên trái, bắt đầu di chuyển sang phải rồi theo hình xoắn ốc vào trong, các giá trị ở các ô chứa số sẽ xuất hiện theo thứ tự 1, 2, ..., k

Hình dưới đây là ví dụ về một bảng số 2 - đẹp kích thước 4x4:



Cho 2 số tự nhiên n và k, vẽ bảng số k - đẹp kích thước n x n

**Input.** Một dòng duy nhất gồm 2 số tự nhiên n và k (2 ≤ n ≤ 200, 2 ≤ k ≤ min (n, 3))

**Ouput.**

* Nếu không có lời giải, chỉ ghi -1.
* Nếu có lời giải, in ra bất kỳ lời giải nào hợp lệ:
* Ghi ra n dòng của bảng số tìm được, mỗi dòng gồm n số, các số cách nhau đúng 1 dấu cách
* Những ô trống không có số thì ô đó nhận giá trị là 0

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 2 | 0 0 1 2 2 1 0 0 1 0 2 0 0 2 0 1 |

**Chú ý.** Đây chỉ là một trong những đáp án hợp lệ. Học sinh hoàn toàn có thể có kết quả khác, miễn thoả mãn yêu cầu đề bài

**Subtask**

* **Subtask 1 (40%):** k = 2
* **Subtask 2 (60%):** k = 3

**BÀI 5. QUÂN XE**

Trong giờ giải lao trên lớp, Alice nghĩ ra một trò chơi để đố Bob. Alice vẽ ra giấy trắng một bảng kích thước 8 × 8, các hàng được đánh số từ 1 đến 8 từ trên xuống dưới, và các cột được đánh số từ 1 đến 8 từ trái sang phải. Alice đã bí mật tô một số ô thành màu đen, các ô còn lại là màu trắng. Biết rằng số ô màu đen nhiều hơn số ô màu trắng. Alice yêu cầu Bob cần phải tìm ra một tập 5 ô đen sao cho nếu đặt 5 quân xe vào đó thì: không có hai quân xe nào nằm cùng hàng hoặc cùng cột.

Bob không được nhìn bảng; thay vào đó, Bob được phép hỏi nhiều lần Alice: trong một tập các ô có bao nhiêu ô trong đó là màu đen. Alice luôn trả lời đúng theo cấu hình bí mật. Vì giờ giải lao rất ngắn nên Bob cần phải đặt càng ít câu hỏi càng tốt để tìm ra một tập 5 ô đen hợp lệ.

**Input**

Dòng đầu chứa số nguyên dương T(1 ≤ *T* ≤ 100) là số bộ test. Dưới đây là mô tả của các trường hợp test.

**Tương tác**

* Để thực hiện phép hỏi, bạn cần chọn ra k cặp số nguyên (r1, c1), (r2, c2), … , (rk, ck) là toạ độ của các ô trong tập các ô cần hỏi và in ra luồng dữ liệu chuẩn một dòng dưới dạng sau: "? k r1 c1 r2 c2 … rk ck".
* Ngay sau đó bạn đọc từ luồng dữ liệu chuẩn một số nguyên là số lượng ô màu đen trong tập ô đã chọn.
* Khi đã tìm được 5 ô đen có toạ độ (x1, y1), (x2, y2), (x3, y3), (x4, y4), (x5, y5) theo yêu cầu, bạn in ra luồng dữ liệu chuẩn một dòng dưới dạng sau:

"! x1, y1, x2, y2, x3, y3, x4, y4, x5, y5".

* Sau khi in ra, đừng quên xuất ký tự xuống dòng (end of line) và làm sạch bộ đệm đầu ra (flush the output). Nếu không, bài làm sẽ bị chấm lỗi. Để làm điều này, hãy sử dụng:
* fflush(stdout) hoặc cout.flush() trong C++
* System.out.flush() trong Java
* flush(output) trong Pascal
* stdout.flush() trong Python

**Điểm số**

Với mỗi test, nếu bạn trả lời đúng tập các điểm cần tìm, gọi *Q* là số lần thực hiện phép hỏi lớn nhất trong toàn bộ các trường hợp test. Điểm cho mỗi test được tính như sau:

* 0 điểm nếu Q > 64;
* 100% số điểm nếu Q ≤ 9;
* Ngược lại bạn nhận được 0.9 − (Q − 9) × 0.01.

**Subtask**

* **Subtask 1 (30 điểm):** Có đúng 5 ô được tô màu đen trong số các ô trên đường chéo chính (các ô có chỉ số hàng bằng chỉ số cột).
* **Subtask 2 (70 điểm):** Không có ràng buộc gì thêm.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2  0  3 | ? 3 1 1 2 2 3 3  ? 1 3 3  ? 3 4 4 5 5 6 6  ! 1 1 2 2 4 4 5 5 6 6 |

**Giới hạn thời gian:** 1.0s

**Giới hạn bộ nhớ:** 500M

**\_\_\_\_\_\_\_HẾT\_\_\_\_\_\_\_**

**LINK SUBMIT**

Các bạn submit tại đường dẫn sau: <https://hnoj.edu.vn/contest/25bc> (Mã: thayson)