**Street Parade**

**Link submit:** <https://www.spoj.com/problems/STPAR/>

**Solution:**

|  |  |
| --- | --- |
| C++ | <https://ideone.com/WmzZAV> |
| Java | <https://ideone.com/MRbIkE> |
| Python | <https://ideone.com/OE6SDw> |

**Tóm tắt đề:**

Ban tổ chức muốn sắp xếp các chiếc xe đang đỗ trong bãi ra trình diễn theo thứ tự từ nhỏ đến lớn. Cho rằng một chiếc xe có quyền đỗ tạm vào một con hẻm trên đường đi nếu chưa tới lượt nó và diện tích mỗi con đường chỉ có thể cho phép tối đa một chiếc xe di chuyển tại một thời điểm.

Nhiệm vụ của bạn là xác định xem ban tổ chức có thể thực hiện được màn trình diễn đúng như dự định hay không.

**Input:**

Gồm một hoặc nhiều bộ test. Mỗi bộ test có định dạng như sau:

* Dòng đầu tiên chứa một số nguyên dương n (n ≤ 1.000) – số lượng xe đang đỗ trong bãi.
* Dòng thứ hai chứa n số nguyên từ 1 đến n với thứ tự ngẫu nhiên đại diện cho số thứ tự của các chiếc xe đang đỗ trong bãi.

Kết thúc toàn bộ bộ test là số 0.

**Output:**

Với mỗi bộ test, in ra trên một dòng là “yes” nếu ban tổ chức có thể thực hiện được màn trình diễn đúng thứ tự, ngược lại in ra “no”.

**Ví dụ:**

|  |  |
| --- | --- |
| 5  5 1 2 4 3  0 | yes |

**Giải thích ví dụ:**

Ví dụ trên chỉ gồm 1 bộ test.

Thứ tự trình diễn được mong muốn là {1, 2, 3, 4, 5}.

Trong bãi đậu xe hiện có các chiếc xe với thứ tự {5, 1, 2, 4, 3}.

* Thứ tự được yêu cầu là 1. Xe số 5 không đúng thứ tự trình diễn nên đưa vào hẻm.
* Thứ tự được yêu cầu là 1. Xe số 1 đúng với thứ tự nên sẽ ra trình diễn.
* Thứ tự được yêu cầu là 2. Xe số 2 cũng đúng thứ tự nên sẽ ra trình diễn.
* Thứ tự được yêu cầu là 3. Xe số 4 không đúng với thứ tự nên được đưa vào hẻm.
* Thứ tự được yêu cầu là 4. Xe số 3 đúng với thứ tự, ta đưa ra trình diễn.
* Thứ tự được yêu cầu là 4. Từ trong hẻm, lúc này xe số 4 có thể đi ra trình diễn.
* Thứ tự được yêu cầu là 5. Từ trong hẻm, lúc này xe số 5 có thể đi ra trình diễn.

Như vậy, bằng cách chuyển qua chuyển lại giữa các xe trong bãi và hẻm, ta có thể thực hiện màn trình diễn đúng với thứ tự đã định. Do đó in ra “yes”.

**\* Lưu ý:**

* Xe từ bãi đậu xe có thể đi ra chỗ trình diễn hoặc có thể chuyển vào trong hẻm. Nhưng xe đã vào hẻm rồi thì không được trở lại bãi đậu xe.
* Xe đưa ra trình diễn có thể đi từ một trong hai vị trí: từ bãi đậu xe hoặc từ đầu hẻm.

**Hướng dẫn giải:**

Nhận xét:

* Ta có thể mô phỏng hoạt động của con hẻm bằng CTDL stack với nguyên lý xe nào vào hẻm sau thì có thể đi ra trước.
* Các chiếc xe được trình diễn theo số thứ tự từ nhỏ đến lớn, tức từ xe có số thứ tự 1 đến số thứ tự n. Do đó, ta hoàn toàn có thể biết được chiếc xe hiện tại đang cần đưa ra trình diễn có số thứ tự là bao nhiêu bằng cách sử dụng một biến đếm.
* Như vậy, với mỗi lần cần đưa một chiếc xe ra trình diễn, ta đối chiếu số thứ tự của xe được yêu cầu với xe ở bãi đỗ và xe ở hẻm để quyết định đưa xe nào ra trình diễn. Trong trường hợp không có xe nào được đem ra, ta cần đưa xe đang đỗ trong bãi vào hẻm để xét xe kế tiếp nằm trong bãi.

Ta có các bước giải bài này như sau:

* Bước 1: Đọc thông tin các xe hiện có trong bãi vào một mảng.
* Bước 2: Khởi tạo một stack dùng để mô phỏng con hẻm. Ban đầu stack rỗng.
* Bước 3: Sử dụng một biến đếm lưu số thứ tự xe cần để đem ra trình diễn với giá trị khởi tạo bằng 1.
* Bước 4: Duyệt qua từng chiếc xe trong bãi:
  + Nếu số thứ tự đang cần đúng bằng số thứ tự xe trong bãi đỗ xe thì đưa xe này ra trình diễn và tăng thứ tự trình diễn.
  + Ngược lại, xét xem xe trong hẻm có phù hợp không:
    - Nếu có, đem xe trong hẻm ra trình diễn và tăng thứ tự trình diễn.
    - Nếu không, nghĩa là cả xe đang nằm trong bãi và xe đầu hẻm đều không đúng thứ tự, ta đưa xe hiện tại trong bãi vào hẻm để xét xe kế tiếp.
* Bước 5: Nếu hết xe trong bãi đỗ, ta cần kiểm tra các xe còn lại trong hẻm có thỏa thứ tự còn lại hay không và đưa ra trình diễn nếu đúng.
* Bước 6: Kiểm tra biến lưu thứ tự trình diễn. Nếu biến đạt tới thứ tự (n + 1) nghĩa là ta đã đem đủ n chiếc xe ra trình diễn, in “yes”. Ngược lại, in “no”.
* Bước 7: Reset các biến cần thiết, chuẩn bị cho bộ test tiếp theo.

**Độ phức tạp:** **O(T \* n)** với T là số lượng bộ test và n là số lượng xe hiện có trong bãi đỗ.