**Array**

**Link submit:** <http://codeforces.com/problemset/problem/224/B>

**Solution:**

|  |  |
| --- | --- |
| C++ | <https://ideone.com/kIMZ5d> |
| Java | <https://ideone.com/mMXMq7> |
| Python | <https://ideone.com/82FtWn> |

**Tóm tắt đề:**

Cho mảng a gồm n phần tử có vị trí từ 1 đến n và một con số k.

Nhiệm vụ của bạn là tìm ra một đoạn [l, r] sao cho nó chứa đúng k phần tử phân biệt. Lưu ý rằng không nhất thiết đoạn tìm được phải là đoạn có chiều dài nhỏ nhất thỏa yêu cầu, tuy nhiên nó không được phép chứa bất cứ đoạn con nào cũng chứa đúng k phần tử phân biệt bên trong nó. Nói cách khác, không có đoạn [x, y] nào, với 1 ≤ l ≤ x ≤ y ≤ r ≤ n, khác [l, r] thỏa mãn tính chất trên.

**Input:**

Dòng đầu tiên chứa hai số nguyên *n* và *k* (1 ≤ n, k ≤ 105).

Dòng tiếp theo chứa n số nguyên *ai* (1 ≤ ai ≤ 105) là phần tử thứ i trong mảng a.

**Output:**

In ra trên một dòng đoạn cần tìm xác định bởi cặp l, r.

Nếu không có đoạn nào thỏa yêu cầu, in ra cặp “-1 -1”. Nếu có nhiều đáp án thì in ra một đáp án bất kỳ trong số đó.

**Ví dụ:**

|  |  |
| --- | --- |
| 4 2 1 2 2 3 | 1 2 |

|  |  |
| --- | --- |
| 8 3 1 1 2 2 3 3 4 5 | 2 5 |

|  |  |
| --- | --- |
| 7 4 4 7 7 4 7 4 7 | -1 -1 |

**Giải thích ví dụ:**

***Ví dụ 1:*** Đoạn cần tìm phải chứa đúng 2 phần tử phân biệt. Có các đoạn thỏa yêu cầu này là:

* Đoạn [1, 2] có [l, r] = [1, 2] chứa 2 số phân biệt là {1, 2}.
* Đoạn [2, 3] có [l, r] = [3, 4] chứa 2 số phân biệt là {2, 3}.

Đoạn [1, 2, 2] tuy cũng chứa 2 số phân biệt nhưng nó lại có đoạn con [1, 2] cũng thỏa yêu cầu này nên ta không chọn. Tương tự với đoạn [2, 2, 3]. Do đó, đáp án của ta chỉ có thể là "1 2" hoặc "3 4".

***Ví dụ 2:*** Đoạn cần tìm phải chứa đúng 3 phần tử phân biệt. Có các đoạn thỏa yêu cầu này là:

* Đoạn [1, 2, 2, 3] có [l, r] = [2, 5] chứa 3 số phân biệt là {1, 2, 3}.
* Đoạn [2, 3, 3, 4] có [l, r] = [4, 7] chứa 3 số phân biệt là {2, 3, 4}.
* Đoạn [3, 4, 5] có [l, r] = [6, 8] chứa 3 số phân biệt là {3, 4, 5}.

Đoạn [1, 1, 2, 2, 3] tuy cũng chứa 3 số phân biệt nhưng nó lại có đoạn con [1, 2, 2, 3] cũng thỏa yêu cầu này nên ta không chọn. Tương tự với đoạn [2, 2, 3, 3, 4] hay [1, 1, 2, 2, 3, 3]... Do đó, đáp án của ta chỉ có thể là "2 5", "4 7" hoặc "6 8".

***Ví dụ 3:*** Đoạn cần tìm phải chứa đúng 4 phần tử phân biệt. Tuy nhiên ta thấy rằng không có bất cứ đoạn nào thỏa yêu cầu này, thậm chí nếu đoạn được chọn là toàn bộ mảng thì cũng chỉ được 2 phần tử phân biệt là {4, 7}. Do đó, ta in ra "-1 -1".

**Hướng dẫn giải:**

Ý tưởng giải dựa trên kỹ thuật Two Pointers với hai biến chạy i và j:

* Bước 1: Sử dụng biến chạy i để chạy tìm đoạn đầu tiên chứa đủ k số phân biệt.
* Bước 2: Sử dụng biến chạy j chạy từ dưới lên nhằm rút ngắn đoạn tìm được, đảm bảo trong [j, i] không còn bất cứ đoạn con nào cũng thỏa yêu cầu.
* Bước 3: In ra kết quả bài toán nằm trong hai biến j và i.

Trong đó, để kiểm soát số phần tử phân biệt đã tìm được, ta dùng thêm một mảng đếm phân phối fre[] với fre[n] là số lần n xuất hiện trong khoảng [j, i]. Với mỗi lần fre[n] thay đổi từ 0 lên 1, tức là ta vừa nhận được thêm một số mới không trùng với những số trước đó, ta tăng biến đếm số lượng phần tử phân biệt lên 1.

**Độ phức tạp:** **O(n)** với n số lượng phần tử trong dãy.