

## Tìm kiếm nhị phân

Xét một thủ tục tìm kiếm nhị phân sau:

```
int binary_search(vector<int> &a, int value) {
    int l = 0;
    int r = (int)a.size() - 1;
    while (l <= r) {
        int m = (l + r + 1) / 2;
        if (a[m] == value)
            return m;
        else if (a[m] > value)
            r = m - 1;
        else
            l = m + 1;
    }
    return -1;
}
```

Thuật toán tìm kiếm nhị phân chỉ đúng khi `vector` `a` đã được sắp xếp tăng dần. Tuy nhiên, đôi khi `a` không được sắp xếp tăng dần nhưng thuật toán tìm kiếm nhị phân trên vẫn hoạt động đúng như kỳ vọng.

**Yêu cầu:** Cho số nguyên dương  $N$  và giá trị  $v$ . Đếm số lượng `vector` `a` là một hoán vị của  $N$  số nguyên dương từ 1 tới  $N$  thỏa mãn: khi thủ tục `binary_search(a, v)` được gọi, thủ tục sẽ trả được về chỉ số  $i$  mà  $a[i] = v$ . Vì số lượng hoán vị thỏa mãn có thể rất lớn, bạn chỉ cần in ra phần dư của nó sau khi chia cho  $10^9+7$ .

**Input:** đọc từ file **binsearch.in**

- Gồm một dòng chứa hai số nguyên dương  $N$  và  $v$
- $1 \leq N \leq 10000$
- $1 \leq v \leq N$

**Output:** ghi ra file **binsearch.out**

In ra số lượng hoán vị modulo  $10^9 + 7$ .

**Subtask:**

*Subtask 1 (10%):*  $N \leq 10$

*Subtask 2 (10%):*  $N \leq 20$

*Subtask 3 (30%):*  $N \leq 100$

*Subtask 4 (25%):*  $N \leq 1000$

*Subtask 5 (25%):*  $N \leq 10000$

**Ví dụ:**

binsearch.in	binsearch.out
--------------	---------------

3 1	4
5 2	66
7 1	2160
9 4	135360