

# Problem 526: Beautiful Arrangement

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 64.67%

**Paid Only:** No

**Tags:** Array, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask

## Problem Description

Suppose you have `n` integers labeled `1` through `n`. A permutation of those `n` integers `perm` (\*\*1-indexed\*\*) is considered a \*\*beautiful arrangement\*\* if for every `i` ( $1 \leq i \leq n$ ), \*\*either\*\* of the following is true:

- \* `perm[i]` is divisible by `i`.
- \* `i` is divisible by `perm[i]`.

Given an integer `n`, return \_the\*\*number\*\* of the \*\*beautiful arrangements\*\* that you can construct\_.

**Example 1:**

**Input:** n = 2   **Output:** 2   **Explanation:** The first beautiful arrangement is [1,2]: - perm[1] = 1 is divisible by i = 1 - perm[2] = 2 is divisible by i = 2 The second beautiful arrangement is [2,1]: - perm[1] = 2 is divisible by i = 1 - i = 2 is divisible by perm[2] = 1

**Example 2:**

**Input:** n = 1   **Output:** 1

**Constraints:**

\*  $1 \leq n \leq 15$

## Code Snippets

**C++:**

```
class Solution {  
public:  
    int countArrangement(int n) {  
  
    }  
};
```

**Java:**

```
class Solution {  
public int countArrangement(int n) {  
  
}  
}
```

**Python3:**

```
class Solution:  
    def countArrangement(self, n: int) -> int:
```