

# Problem 544: Output Contest Matches

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 77.44%

**Paid Only:** Yes

**Tags:** String, Recursion, Simulation

## Problem Description

During the NBA playoffs, we always set the rather strong team to play with the rather weak team, like making the rank `1` team play with the rank `nth` team, which is a good strategy to make the contest more interesting.

Given `n` teams, return \_their final contest matches in the form of a string\_.

The `n` teams are labeled from `1` to `n`, which represents their initial rank (i.e., Rank `1` is the strongest team and Rank `n` is the weakest team).

We will use parentheses `(`, and `)` and commas `,` to represent the contest team pairing. We use the parentheses for pairing and the commas for partition. During the pairing process in each round, you always need to follow the strategy of making the rather strong one pair with the rather weak one.

**Example 1:**

**Input:** `n = 4` **Output:** `"((1,4),(2,3))"` **Explanation:** In the first round, we pair the team 1 and 4, the teams 2 and 3 together, as we need to make the strong team and weak team together. And we got (1, 4),(2, 3). In the second round, the winners of (1, 4) and (2, 3) need to play again to generate the final winner, so you need to add the parentheses outside them. And we got the final answer ((1,4),(2,3)).

**Example 2:**

**Input:** `n = 8` **Output:** `"(((1,8),(4,5)),((2,7),(3,6)))"` **Explanation:** First round: (1, 8),(2, 7),(3, 6),(4, 5) Second round: ((1, 8),(4, 5)),((2, 7),(3, 6)) Third round: (((1, 8),(4, 5)),((2, 7),(3, 6)))

6))) Since the third round will generate the final winner, you need to output the answer (((1,8),(4,5)),((2,7),(3,6))).

**\*\*Constraints:\*\***

\* `n == 2x` where `x` in in the range `[1, 12]`.

## Code Snippets

### C++:

```
class Solution {
public:
    string findContestMatch(int n) {

    }
};
```

### Java:

```
class Solution {
    public String findContestMatch(int n) {

    }
}
```

### Python3:

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