# 100340. Maximum Height of a Triangle

| | |
|---|---|
| User Accepted: | 8421 |
| User Tried: | 12450 |
| Total Accepted: | 8552 |
| Total Submissions: | 23035 |
| Difficulty: | Easy |

You are given two integers `red` and `blue` representing the count of red and blue colored balls. You have to arrange these balls to form a triangle such that the 1st row will have 1 ball, the 2nd row will have 2 balls, the 3rd row will have 3 balls, and so on.

All the balls in a particular row should be the **same** color, and adjacent rows should have **different** colors.

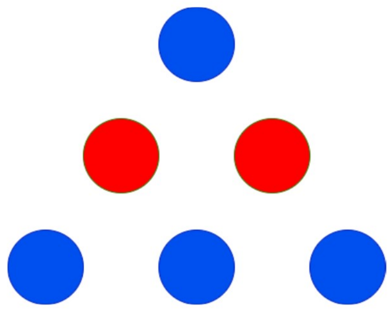
Return the **maximum** *height of the triangle* that can be achieved.

**Example 1:**

**Input:** red = 2, blue = 4

**Output:** 3

**Explanation:**



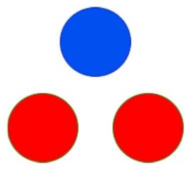
The only possible arrangement is shown above.

**Example 2:**

**Input:** red = 2, blue = 1

**Output:** 2

**Explanation:**



The only possible arrangement is shown above.

**Example 3:**

**Input:** red = 1, blue = 1

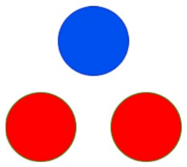
**Output:** 1

**Example 4:**

**Input:** red = 10, blue = 1

**Output:** 2

**Explanation:**



The only possible arrangement is shown above.

**Constraints:**

- `1 <= red, blue <= 100`

C++

```
#import <cmath>

class Solution {
public:
    int maxHeightOfTriangle(const int red, const int blue) {
        return(max(attempt(red, blue), attempt(blue, red)));

    }

private:

    int attempt(const int color1, const int color2){
        if (abs(heightOfEvens(color1) - heightOfOdds(color2)) == 1){
            return(max(heightOfEvens(color1), heightOfOdds(color2)));
        }
        else {
            return(min(heightOfEvens(color1), heightOfOdds(color2))+1);
        }
    }

    int heightOfEvens(const int color){
        return find_height(color, 2);
    }

    int heightOfOdds(const int color){
        return find_height(color, 1);
    }

    int find_height(const int color, int start) {
        int subtractor = start;
        int num = color;

        while (num >= subtractor) {
            num -= subtractor;
            subtractor += 2;
        }
        subtractor -= 2;

    return subtractor;
}
};
```

☑ **Custom Testcase** Use Example Testcases

```
2
4
10
1
```

How to create a testcase

Run Submit

Submission Result: Accepted (/submissions/detail/1304371509/) More Details (/submissions/detail/1304371509/)

Share your acceptance!