

# 题目

给你两个正整数  $x$  和  $y$ ，分别表示价值为 75 和 10 的硬币的数目。

Alice 和 Bob 正在玩一个游戏。每一轮中，Alice 先进行操作，Bob 后操作。每次操作中，玩家需要拿出价值总和为 115 的硬币。如果一名玩家无法执行此操作，那么这名玩家输掉游戏。

两名玩家都采取最优策略，请你返回游戏的赢家。

## 举例：

示例 1：

输入：  $x = 2, y = 7$

输出： "Alice"

解释：

游戏一次操作后结束：

Alice 拿走 1 枚价值为 75 的硬币和 4 枚价值为 10 的硬币。

示例 2：

输入：  $x = 4, y = 11$

输出： "Bob"

解释：

游戏 2 次操作后结束：

Alice 拿走 1 枚价值为 75 的硬币和 4 枚价值为 10 的硬币。

Bob 拿走 1 枚价值为 75 的硬币和 4 枚价值为 10 的硬币。

## 提示：

$1 \leq x, y \leq 100$

# 思路

因为 10 的倍数不可能等于 115，所以面额为 10 的硬币不能单独选，至少要选 1 个面额为 75 的硬币。

又由于  $75 \cdot 2 = 150 > 115$ ，所以面额为 75 的硬币要恰好选 1 个。

由于  $75 + 10 \cdot 4 = 115$ ，所以面额为 10 的硬币要恰好选 4 个。

本质上来说，我们在求解二元一次不定方程  $75a + 10b = 115$ ，它有唯一正整数解  $a=1, b=4$ 。

如果一开始 Alice 就没法选，或者偶数轮后 Alice 没法选，那么 Bob 胜出，否则 Alice 胜出。

## 优化

设  $k = \min(x, \lfloor y/4 \rfloor)$ ，这是能玩的回合数，判断 k 的奇偶性即可。

# 代码

C++:

```
class Solution {
public:
    string losingPlayer(int x, int y) {
        return min(x, y / 4) % 2 ? "Alice" : "Bob";
    }
};
```