370A - Rook, Bishop and King

There are two approaches to this task. The first is use BFS to find the shortest path three times. The second is to notice that:

- A rook can reach the destination in one or two moves. If the starting and the destination fields are in the same row or column, one move is enough.
- A bishop can reach only fields that are colored the same as the starting cell, and can do this in at most two moves: if the starting and the destination fields are on the same diagonal, one move is enough. To find this out, check that r_1 - $c_1 = r_2$ - c_2 OR $r_1 + c_1 = r_2 + c_2$.
- A king should make $max(|r_1$ - $r_2|, |c_1$ - $c_2|)$ moves.

```
int r1, c1, r2, c2;
cin >> r1 >> c1 >> r2 >> c2;
if (r1 == r2 || c1 == c2) cout << 1; else cout << 2;
cout << " ";
if ((r1 + c1) % 2 != (r2 + c2) % 2) cout << 0; else {
    if (r1 + c1 == r2 + c2 || r1 - c1 == r2 - c2) cout << 1; else cout << 2;
}
cout << " ";
cout << max(abs(r1 - r2), abs(c1 - c2)) << endl;
```