

```

// C++ implementation of disjoint set
#include <iostream>
using namespace std;
class DisjSet {
    int *rank, *parent, n;

public:
    // Constructor to create and
    // initialize sets of n items
    DisjSet(int n)
    {
        rank = new int[n];
        parent = new int[n];
        this->n = n;
        makeSet();
    }

    // Creates n single item sets
    void makeSet()
    {
        for (int i = 0; i < n; i++) {
            parent[i] = i;
        }
    }

    // Finds set of given item x
    int find(int x)
    {
        // Finds the representative of the set
        // that x is an element of
        if (parent[x] != x) {

            // if x is not the parent of itself
            // Then x is not the representative of
            // his set,
            parent[x] = find(parent[x]);

            // so we recursively call Find on its parent
            // and move i's node directly under the
            // representative of this set
        }

        return parent[x];
    }

    // Do union of two sets represented
    // by x and y.
    void Union(int x, int y)
    {
        // Find current sets of x and y
        int xset = find(x);
        int yset = find(y);

        // If they are already in same set
        if (xset == yset)
            return;
    }
}

```

```

        // Put smaller ranked item under
        // bigger ranked item if ranks are
        // different
        if (rank[xset] < rank[yset]) {
            parent[xset] = yset;
        }
        else if (rank[xset] > rank[yset]) {
            parent[yset] = xset;
        }

        // If ranks are same, then increment
        // rank.
        else {
            parent[yset] = xset;
            rank[xset] = rank[xset] + 1;
        }
    }
};

int main()
{
    DisjSet obj(5);
    obj.Union(0, 2);
    obj.Union(4, 2);
    obj.Union(3, 1);
    if (obj.find(4) == obj.find(0))
        cout << "Yes\n";
    else
        cout << "No\n";
    if (obj.find(1) == obj.find(0))
        cout << "Yes\n";
    else
        cout << "No\n";

    return 0;
}

```