Write a C++ code for Exhaustive search

using namespace std;

int maxPackedSets(vector<int>& items,

                  vector<set<int> >& sets)

{

  // Initialize the maximum number of sets that can be

  // packed to 0

  int maxSets = 0;

  // Loop through all the sets

  for (auto set : sets) {

    // Initialize the number of sets that can be packed

    // to 0

    int numSets = 0;

    // Loop through all the items

    for (auto item : items) {

      // Check if the current item is in the current

      // set

      if (set.count(item)) {

        // If the item is in the set, increment

        // the number of sets that can be packed

        numSets += 1;

        // Remove the item from the set of items,

        // so that it is not counted again

        items.erase(remove(items.begin(),

                           items.end(), item),

                    items.end());

      }

    }

    // Update the maximum number of sets that can be

    // packed

    maxSets = max(maxSets, numSets+1);

  }

  return maxSets;

}

int main()

{

  // Set of items

  vector<int> items = { 1, 2, 3, 4, 5, 6 };

  // List of sets

  vector<set<int> > sets

    = { { 1, 2, 3 }, { 4, 5 }, { 5, 6 }, { 1, 4 } };

  // Find the maximum number of sets that

  // can be packed into the given set of items

  int maxSets

    = maxPackedSets(items, sets);

  // Print the result

  cout << "Maximum number of sets that can be packed: "

    << maxSets << endl;

  return 0;

}