```cpp
/*
CodeLovers

Nguyen, Da
Banh, Alex
Ton, An

CS A250
April 28, 2018

Lab 11
*/

#include <iostream>
#include <set>
#include <map>

using namespace std;

int countClumps(const multiset<int>& myset);
bool linearIn(const multimap<int, int>& mymap);
void multiples(set<int>& myset, int n);
void printSet(const set<int>& myset);

int main()
{
    multiset<int> mset1 = { 1, 2, 2, 3, 4, 4 };
    multiset<int> mset2 = { 1, 2, 2, 3 };
    multiset<int> mset3 = {};
    cout << "The number of clumps in each set:\n" <<
        "set 1 -> " << countClumps(mset1) <<
        "\nset 2 -> " << countClumps(mset2) <<
        "\nset 3 -> " << countClumps(mset3);
    multimap<int, int> map1 = { { 3, 3 }, { 4, 4 }, { 5, 1 }, { 6, 2 }, { 7, 3 }, ↵
       { 7, 5 }
    };
    multimap<int, int> map2 = { { 3, 6 }, { 4, 6 }, { 5, 6 }, { 6, 6 } };
    multimap<int, int> map3 = { { 3, 3 }, { 3, 3 }, { 3, 4 }, { 4, 3 } };
    cout << "\n\nCheck if the map is linear\n" <<
        "map 1 -> " << (linearIn(map1) ? "True" : "False") <<
        "\nmap 2 -> " << (linearIn(map2) ? "True" : "False") <<
        "\nmap 3 -> " << (linearIn(map3) ? "True" : "False");
    set<int> set1;
    set<int> set2;
    set<int> set3;
    cout << "\n\nSet of 10 multiples of 2: ";
    multiples(set1, 2);
    printSet(set1);
    cout << "\nSet of 10 multiples of 5: ";
    multiples(set2, 5);
    printSet(set2);
    cout << "\nSet of 10 mutiples of 15: ";
```

```
52      multiples(set3, 15);
53      printSet(set3);
54      cout << "\n";
55      system("Pause");
56      return 0;
57  }
58
59  int countClumps(const multiset<int>& myset)
60  {
61      int count = 0;
62      multiset<int> newSet;
63      for (multiset<int>::const_iterator iterSet = myset.cbegin();
64          iterSet != myset.cend(); iterSet++)
65      {
66          if (myset.count(*iterSet) >= 2 && (newSet.find(*iterSet) == newSet.end
              ()))
67              count++;
68          newSet.insert(*iterSet);
69      }
70      return count;
71  }
72
73  bool linearIn(const multimap<int,int>& mymap)
74  {
75      for (multimap<int, int>::const_iterator iterMap = mymap.cbegin();
76          iterMap != mymap.cend(); iterMap++)
77      {
78          pair<multimap<int, int>::const_iterator, multimap<int,
              int>::const_iterator>
79              newPair = mymap.equal_range(iterMap->second);
80          if (distance(newPair.first, newPair.second) == 0) return false;
81      }
82      return true;
83  }
84
85  void multiples(set<int>& myset, int n)
86  {
87      int count = 2;
88      while (myset.size() < 10)
89      {
90          if (n*count % 10 != n)
91          {
92              myset.insert(n * count);
93          }
94          count++;
95      }
96  }
97
98  void printSet(const set<int>& myset)
99  {
100     for (set<int>::iterator iterSet = myset.begin(); iterSet != myset.end();
          iterSet++)
```

```
101     {
102         cout << *iterSet << " ";
103     }
104 }
```