```
#include <iostream>
#include <string>
#include <tr1/unordered_map> //for unordered_map
using std::tr1::unordered_map; //for unordered_map
using namespace std;

int main()
{
  unordered_map<string, int> mylist; //unordered map

  string names[] = {"name1", "name1", "name2", "name3", "name4", "name3", "name5"};
  string key;

  //build an unordered_map <key, count>
  for (int i=0; i<7; i++)
  {
    key = names[i];
    if (mylist.find(key) != mylist.end()) //if key found -- already existing key
      mylist[key]+=1; //increment count
    else //new key
      mylist[key] = 1; //count is 1
  }

  //search unordered_map
  for (int i=0; i<7; i++)
  {
    key = names[i];
    if (mylist.find(key) != mylist.end()) //if key found -- already existing key
      cout<<"key = "<<key<<" ; count = "<<mylist[key]<<endl;
    else //key not found
      cout<<key<<"  not found"<<endl;
  }

return 0;
}


////////////////////////////
/////// Python has "dictionary" for unordered_map in C++
////////////////////////////
////
////  mydictionary = {} ##{} means empty dictionary
////
////  if mydictionary.has_key(myKey):
////     mydictionary[myKey] = int(mydictionary[myKey])+1
////  else:
////     mydictionary[myKey] = 1
////
////////////////////////////
```