#include <set>

#include <iostream>

#include <string>

using namespace std;

template <typename T>

void DisplayContents (const T& container)

{

for (auto iElement = container.cbegin();

iElement != container.cend();

++ iElement)

cout << \*iElement << endl;

cout << endl;

}

struct ContactItem

{

string name;

string phoneNum;

string displayAs;

ContactItem(const string& nameInit, const string & phone)

{

name = nameInit;

phoneNum = phone;

displayAs = (name + ":" + phoneNum);

}

bool operator == (const ContactItem& itemToCompare) const

{

return (itemToCompare.name == this->name);

}

bool operator < (const ContactItem& itemToCompare) const

{

return (this->name < itemToCompare.name);

}

operator const char\*() const

{

return displayAs.c\_str();

}

};

int main()

{

set<ContactItem> setContacts;

setContacts.insert(ContactItem("Chronis","6943146552"));

setContacts.insert(ContactItem("Dominant","6945647832"));

setContacts.insert(ContactItem("Kotsos","6944324114"));

setContacts.insert(ContactItem("Sophie","6944422666"));

setContacts.insert(ContactItem("Aimilios","6947689001"));

DisplayContents(setContacts);

cout << "Give the name that you want to delete:";

string inputName;

getline(cin,inputName);

auto contactFound = setContacts.find(ContactItem(inputName,""));

if(contactFound != setContacts.end())

{

setContacts.erase(contactFound);

cout << "Content show after delete" << inputName << endl;

DisplayContents(setContacts);

}

else

cout << "Contact is not found" << endl;

return 0;

}