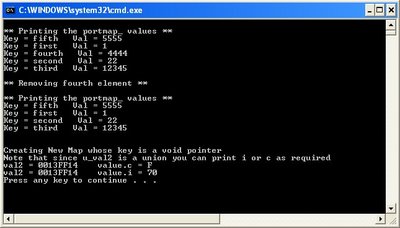
**An example of C++ maps**

The following example shows the working of C++ maps. This also has a simple example of union which is sometimes used in C and C++ programs

//Program tested on Microsoft Visual Studio 2008 - Zahid Ghadialy  
//This program shows an example of how maps work.  
//In maps, the first parameter is key and second value  
//The keys in the map are automatically softed from lower to higher  
#include<iostream>  
#include<map>  
#include<string>  
  
using namespace std;  
  
//defining a union that is used with newMap\_  
union uu  
{  
 char c;  
 int i;  
} u;  
  
//Lets define two different maps  
//The first parameter is key and second value  
map<string, int> portMap\_;  
map<void \*, uu> newMap\_;  
  
int main()  
{  
 //first entry in portmap  
 portMap\_["first"] = 1;  
  
 //example of using the iterator  
 map<string, int>::const\_iterator it;  
 string z = "second";  
 it = portMap\_.find(z); //not in the map so wont be found  
 if(it == portMap\_.end())  
 {  
 portMap\_[z] = 22; //add second element  
 }  
  
 //Add thrid element directly  
 z = "third";  
 portMap\_[z] = 12345;  
  
 //Add 4th element by insert  
 portMap\_.insert(pair<string,int>("fourth", 4444));   
  
 //Add 5th element by insert  
 portMap\_.insert(pair<string,int>("fifth", 5555));   
  
  
 cout<<"\n\*\* Printing the portmap\_ values \*\*"<<endl;  
 for(it = portMap\_.begin(); it != portMap\_.end(); ++it)  
 cout<<"Key = "<<it->first<<" Val = "<<it->second<<endl;  
  
 cout<<"\n\*\* Removing fourth element \*\*"<<endl;  
 z = "fourth";  
 it = portMap\_.find(z);  
 portMap\_.erase(it);  
  
 cout<<"\n\*\* Printing the portmap\_ values \*\*"<<endl;  
 for(it = portMap\_.begin(); it != portMap\_.end(); ++it)  
 cout<<"Key = "<<it->first<<" Val = "<<it->second<<endl;  
  
 //Playing with New Map  
 cout<<"\n\nCreating New Map whose key is a void pointer"<<endl;  
  
 uu u\_val1, u\_val2;  
 void \*val1, \*val2;  
 u\_val1.i = 70, val1 = &u\_val1;  
 newMap\_[val1]=u\_val1;  
  
 val2 = val1;  
 map<void \*, uu>::const\_iterator it\_new;  
 it\_new = newMap\_.find(val2);  
 if(it\_new != newMap\_.end())  
 {  
 u\_val2 = it\_new->second;  
 cout<<"Note that since u\_val2 is a union you can print i or c as required"<<endl;  
 cout<<"val2 = "<<val2<<" value.c = "<<u\_val2.c<<endl;  
 cout<<"val2 = "<<val2<<" value.i = "<<u\_val2.i<<endl;  
 }  
  
 return 0;  
}

The output is as follows:  
[](http://1.bp.blogspot.com/_mAJDMuyu_WI/SbjHlxJ7SoI/AAAAAAAABSA/jnKq1VinsfA/s1600-h/maps1.jpg)