|  |  |  |
| --- | --- | --- |
| Design an application in C++ using STL map, STL algorithms as per given requirements. Add required class and implementation .  Reuse the code used in STL List and STL Stack assignment problem which processes a list of server name and port numbers | | |
| Requirement Tag | Requirement Description | Comments |
| FSTR/01 | Read a list of server name and port numbers (separated by a configurable delimiter) as command line arguments and store in suitable STL list container  [Input may contain duplicate server names entries with same or different port numbers] | |
| FSTR/02 | Display the received objects using containers iterator |  |
| FSTR/03 | Read and store all objects (with unique server names ) in a map1 using name as key and port as value. In case of duplicate entries with same server name, store the last port number. Do count and display the duplicate entries. | |
| FSTR/04 | Display the stored entries in map |  |
| FSTR/05 | Prompt and read server name, search and delete this entry in map if found |  |
| FSTR/06 | Prompt and read server name and port number, search and update this in map |  |
| FSTR/07 | Ensure that there are no memory leaks |  |
|  |  |  |
|  |  |  |

#include<iostream>

#include<string>

#include<list>

#include <stack>

#include <map>

using namespace std;

int main(int argc,char \*argv[])

{

char ch;

string str;

string strg;

list<string> ll;

list<string>:: iterator itr=ll.begin();

for(int i=1; i<argc;i++)

{

str=argv[i];

ll.push\_back(str);

}

itr=ll.begin();

cout<<"here is the list";

while(itr!=ll.end())

{

cout<<"\n"<<\*itr++;

}

itr=ll.begin();

ll.unique();

for (auto ip = ll.begin(); ip != ll.end(); ++ip) {

cout << \*itr << " ";

}

itr=ll.begin();

map<string,int> mp;

map<int,int>::iterator it;

while(itr!=ll.end())

{

strg =\*itr;

mp.insert({strg});

\*itr++;

for (auto& x: mp) {

cout << "server name"<<x.first << "port name" << x.second << '\n';

}

}

string dels;

cout<<"enter the server to be deleted";

cin>>dels;

for(auto it = mp.begin(); it != mp.end(); ) {

if(it->first = dels)

it = mp.erase(it);

else

++it;

}

for(auto& p : c) {

std::cout << p.second << ' ';

}

cout<<"\nenter the server to be updated";

cin>>upds;

cout<<"\nenter the port name to be updated for the same server";

cin>>portn;

for(auto it = mp.begin(); it != mp.end(); ) {

if(it->first = upds)

it = mp.swap(it);

it->second=mp.swap(portn);

else

++it;

}

for (auto& y: mp) {

std::cout << y.first << ": " << y.second << '\n';

}

return 0;

}