Nikeem Dunkelly-Allen, ECE318, Assignment 2

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include <stdlib.h>

using namespace std;

unsigned int hashf(string s)

{ unsigned int h = 987123654;

for (int i = 0; i < s.length(); i += 1)

h = h \* 691 + s[i];

if (h < 0)

h = -h;

return h;

}

struct place

{ int numc;

string sabb;

string name;

int popu;

double area;

double lati;

double longi;

int roadtoint;

double dist;

place(int num, string st, string na, int pop, double a, double lat, double lon, int road, double d)

{ numc=num;

sabb=st;

name=na;

popu=pop;

area=a;

lati=lat;

longi=lon;

roadtoint=road;

dist=d;

}

void print()

{

cout<<numc<< " "<<sabb<<" "<< name<<" "<< popu<< " "<< area<<" "<<lati<< " "<< longi<< " "<<roadtoint

<< " "<< dist<<endl;

}

};

struct Link

{ place \* P;

Link \* next;

Link(place \* Pl = NULL, Link \* n = NULL)

{ P = Pl;

next = n;

}

};

void print\_links(Link \* content /\*, FILE \* \*/)

{

while(content!= NULL /\* && fin.eof()\*/)

{ cout<<content->P->numc<<" "<<content->P->sabb<<" "<<content->P->name<<" "<<content->P->popu<<" "

<<content->P->area<<" "<<content->P->lati<<" "<<content->P->longi<<" "

<<content->P->roadtoint<<" " <<content->P->dist<<endl;

content=content->next;

}

}

place \* find(Link \* ptr, string na, string st /\* FILE \* \*/)

{

while(ptr!=NULL /\* && fin.eof()\*/)

{

if(ptr->P->sabb==st)

return ptr->P;

ptr=ptr->next;

}

return NULL;

}

void find\_all\_names(Link \* ptr,string na /\* FILE \* \*/)

{ Link \* All\_Names = NULL;

unsigned int pos = hashf(na);

//Link \* ptr = hashtable[pos];

while(ptr!=NULL /\*&& fin.eof()\*/)

{ if(ptr->P->name == na)

{ All\_Names=new Link(ptr->P, All\_Names);

pos++;

}

}

//return NuLL;

}

Link \*\* hashtable[30000];

int main()

{

Link \*\* hashtable = new Link \* [30000](); //NULL pointers

ifstream fin;

fin.open("/home/www/class/een318/named-places.txt");

int numc, popu, roadtoint;

string sabb, name;

double area, lati, longi, dist;

string s;

if (fin.fail())

{

cout << "Couldn't open the file\n";

exit(1);

}

else

{

while(getline(fin, s))

{

if(fin.fail())

{

cout<<"Could not open file."<<endl;

break;

}

numc = stoi(s.substr(0,8));

sabb = s.substr(8,2);

//town name

for(int i=10; i<114; i++)

{ if(s[i]==' ' && s[i+1]==' ')

{ name=s.substr(10,i-10);

break;

}

}

for(int i=58; i<66; i++)

{ if(s[i]!=' ')

{ popu = stoi(s.substr(i,66-i));

break;

}

}

for(int i=67; i<80; i++)

{ if(s[i]!= ' ')

{ area = stod(s.substr(i,80-i));

break;

}

}

for(int i=80; i<90; i++)

{ if(s[i]!= ' ')

{ lati = stod(s.substr(i,90-i));

break;

}

}

for(int i= 90; i<101; i++)

{ if(s[i]!=' ')

{ longi = stod(s.substr(i,101-i));

break;

}

}

for(int i=101; i<106; i++)

{ if(s[i]!= ' ')

{ roadtoint = stoi(s.substr(i,106-i));

break;

}

}

for(int i=106; i<=114; i++)

{ if(s[i]!= ' ')

{ dist = stod(s.substr(i, 114-i));

break;

}

}

int const size = 30000;

unsigned int h=hashf(name);

h=h%size;

Link \* ptr = hashtable[h];

while(ptr != NULL)

{ ptr=ptr->next;

}

if(ptr==NULL)

hashtable[h] = new Link(new place(numc, sabb, name, popu, area, lati, longi, roadtoint

, dist));

}

}

fin.close();

while(true)

{

cout << "Welcome to my cool program!" << endl;

cout << "Type command: "<<endl<< " S - All of the information of the desired city/town in said state." <<endl;

cout << endl << "----------------------------------------------"<<endl;

cout << " N - All of the information for all places with the given name in any state"<<endl;

cout << endl << "-----------------------------------------------"<<endl;

cout << " Q - Exit the program" << endl;

cout << endl << "-----------------------------------------------"<< endl;

string command;

string name;

string sabb;

cin>>command;

if(command=="S")

{ cout<<endl;

cout<<"Enter the name of the city."<<endl;

cin.ignore();

getline(cin, name);

cout<<endl;

cout<<"Enter the two letter state abbreviation for the city you entered.(For example: Florida = FL) " <<endl;

cin>>sabb;

unsigned int pos = hashf(name);

pos=pos%30000;

if(hashtable[pos] == NULL)

{ cout<<"The city/town you've entered does not exist."<<endl;

continue;

}

Link \* place = hashtable[pos];

while(place->P->sabb != sabb)

{

place = place->next;

} // keeps moving through until finds the one

print\_links(place);

place->P->print();

}

if(command=="N")

{ cout<<endl;

cout<<"Enter the name of the city/town."<<endl;

cin.ignore();

getline(cin, name);

cout<<endl;

unsigned int pos = hashf(name);

pos=pos%30000;

if(hashtable[pos] == NULL)

{ cout<<"That city/town does not exist."<<endl;

continue;

}

Link \* place = hashtable[pos];

while(place->P->name != name)

{

place = place->next;

}

print\_links(place);

place->P->print();

}

if(command=="Q")

break;

}

return 0;

}